List & Label

The Report Designer

List & Label

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

Neither this manual nor any part of it may be hard copied or stored in any other way (e.g. digitally) without the express written consent of combit GmbH.

All references to personal or company names in this handbook are, as far as could be determined, purely fictitious. Any similarities are unintentional.

1. Introduction

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

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

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

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

1.1. Unit Systems

List & Label can work with the metric or the inch unit system. Which units are available depends on your application and/or operating system settings. Ask your application's vendor for support of a specific unit system. The examples and screenshots in this manual apply to the metric system. Thus, your application may look slightly different. This makes no difference to the functionality, of course.

1.2. **The Designer Interface**
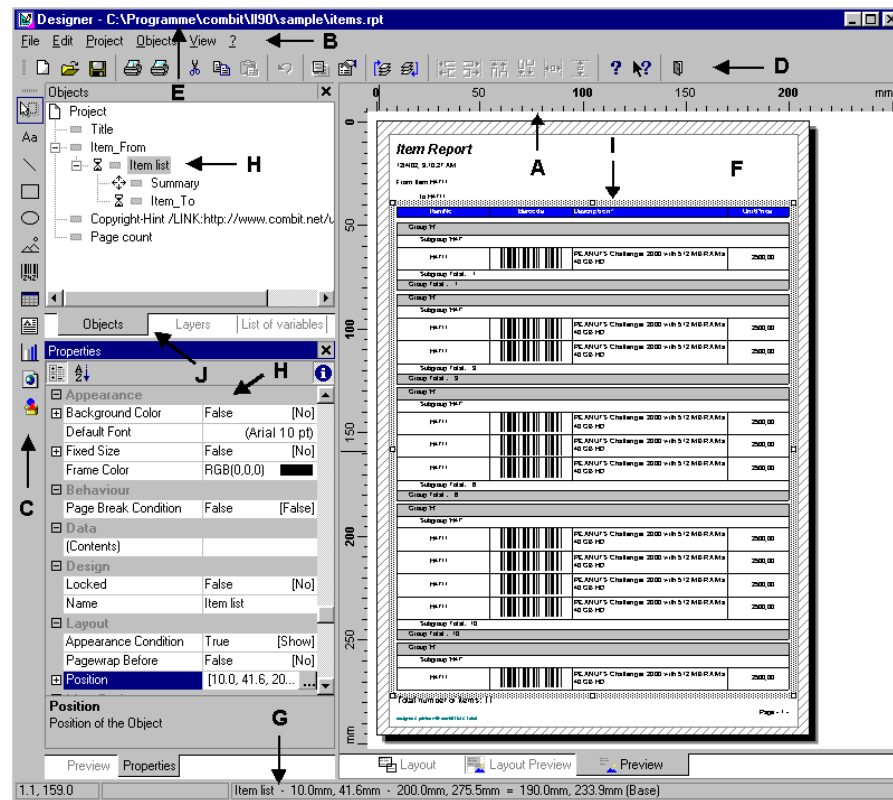
A	Rulers	F	Workspace
B	Menubar	G	Status Line
C	Objectbar	H	Tool Window
D	Toolbar	I	Object
E	Project Title & File Name	J	Tapstrips

1.2.1. Dockable and Floating Tool Windows and Toolbars

Tool Windows

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

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

To do this, select the tool window to be moved by depressing and holding the left mouse button in the title bar. You can now drag the tool window to any desired position. A rectangular frame will be presented which shows the position and size at which the tool window will be placed. If you drag it into the title bar of a previously docked tool window, it will be placed behind the window(s) at this position. If it is dragged into the upper portion of a previously docked window, then the position will be shared and the newly docked window will be in the upper portion. If you drag it into the lower portion of a docked window then the position will be shared and the new window will be in the lower portion. The same is true for the left and right portions of the (docked) tool windows.
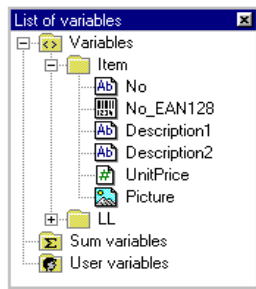
If you place a tool window in a different position, it will not be docked but will be a movable "floating" window.

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

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

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

Toolbars

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

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

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

1.2.2. Object Bar

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

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

Button	Corresponding menu item	Generated object type	Description	Shortcut
	Objects > Select > Selection Mode		Activates the Selection mode for objects in the workspace	CTRL+^
	Objects > Insert > Text	Text	Can contain text or variables / field contents	CTRL+T
	Objects > Insert > Line	Line	For lines	CTRL+L
	Objects > Insert > Rectangle	Rectangle	Can be used as a frame or background	CTRL+R
	Objects > Insert > Ellipse	Ellipse / Circle	For ellipsis or circles	CTRL+I
	Objects > Insert > Picture	Picture	Can contain either a fixed or variable picture	CTRL+D
	Objects > Insert > Barcode	Barcode	For barcodes	CTRL+B

Button	Corresponding menu item	Generated object type	Description	Shortcut
	Objects > Insert > Table	Table	Table objects are used to generate tables or lists. A table consists of the elements headers, data lines, footers, and group lines. The number, width and contents of the columns can be freely defined for every element in a table object	CTRL+E
	Objects > Insert > Formatted Text	Formatted Text object/ RTF	Formatting changes can be made within a line	CTRL+F
	Objects > Insert > Chart	Chart	This object is used for the graphic summary in the form of charts	no shortcut
	Objects > Insert > HTML Text	HTML-Text	With this object, the contents of web sites and other HTML-formatted texts can be displayed	no shortcut
	Objects > Insert > OLE Container	OLE Container	Integration of documents from other applications	no shortcut

1.2.3. Toolbar

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

Button	Corresponding Menu Item
	File > New
	File > Open
	File > Save
	File > Print Sample > Print Sample 1st Page
	File > Print Sample > Print Sample Following Page
	Edit > Cut
	Edit > Copy
	Edit > Paste

Button	Corresponding Menu Item
	Edit > Undo
	Project > Page Setup
	Objects > Contents
	Objects > Arrange > To Front
	Objects > Arrange > To Back
	Objects > Arrange > Alignment > Left
	Objects > Arrange > Alignment > Right
	Objects > Arrange > Alignment > Top
	Objects > Arrange > Alignment > Bottom
	Objects > Arrange > Alignment > Size fit horizontal
	Objects > Arrange > Alignment > Size fit vertical
	Help
	Context Sensitive Help
	File > Exit

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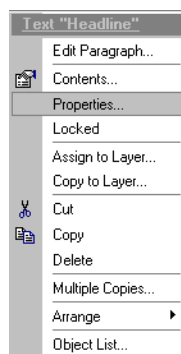
The most important commands that are related to an object can be activated via a context menu. The title bar contains the object's name and type.

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

1.2.5. Status Bar

| 17.4, 24.0 | Select | 2 objects in 2.9mm, 2.2mm - 94.5mm, 31.2mm = 91.6mm, 29.0mm |
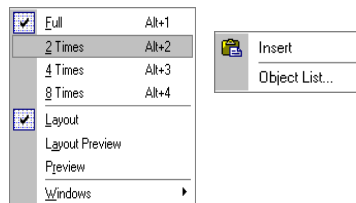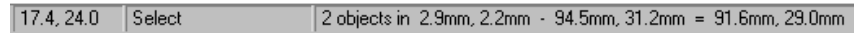
The status bar is divided into three sections:
- the left area indicates the current mouse position in millimeters or inches from the upper left workspace corner.
- the middle area indicates the current mode of operation. (for example, select, draw rectangle, etc.)
- the right area shows the name or type, relative appearance level, as well as the size of a selected object.
- the margin between the object's upper left corner in relation to the same corner of the workspace
- the margin between the object's lower right corner in relation to the same corner of the workspace

1.2.6. Workspace

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

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

Rulers and Scale

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

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

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

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

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

View Modes

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

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

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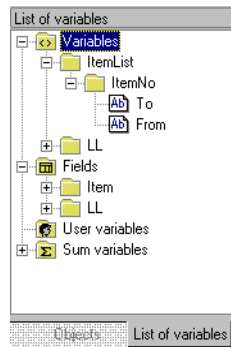
The Variable List displays all variables which are available in the current project; for list type projects, all available fields are additionally displayed.

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

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

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

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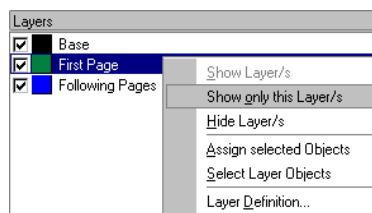
The Layers Window shows the various layers or levels of a project. This window can be repositioned on the workspace at your will.

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

More detailed information on working with layers can be found in Chapter 2.6. Working with Layers.

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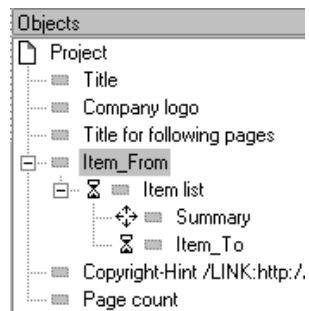
Another tool window is the hierarchical Objects List. All objects in the project are displayed with a small rectangle and it's name in the sequence that they will be printed. The name can be edited by clicking on the current name. The most important commands for the objects are contained in the context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

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

Tip: The special dependence of the objects upon one another, which is very important for printing, is displayed hierarchically. Further, detailed information is contained in Chapter 5.22. Object List.

1.2.10. Preview
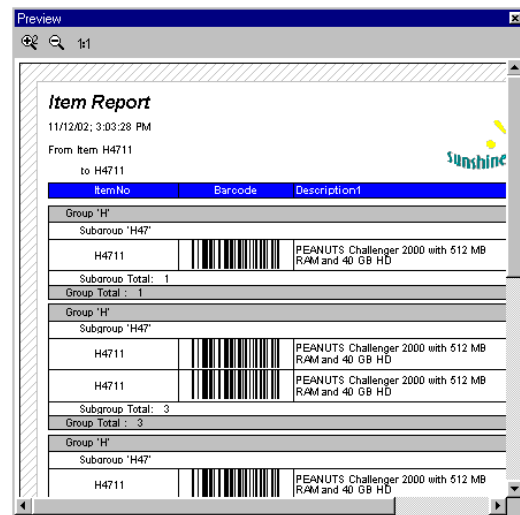
The Preview Window is available in order for you to check the layout of your workspace. Click in the window to change between full size and normal window size. The size of the Preview Window can be changed by dragging the frame. These settings are stored globally for the project type, and are therefore valid for all List & Label projects of the same type.

You can select an area to enlarge by dragging a rectangle. The zoom modes in this preview are independent of the zoom mode in the workspace.

With this button you increase the zoom factor of your current preview by a factor of 2.

This button returns to the previous zoom setting.

Click this button to adapt the zoom factor in order to view the whole page.

1.2.11. Properties Window
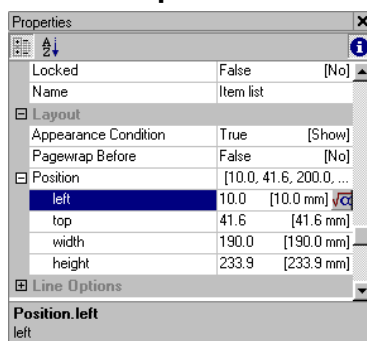
The properties of the individual objects are displayed in this tool window, the Properties Window. The properties are sorted, using the appropriate method, by category. Each category has properties that can be combined into groups. Categories and property groups can be, to improve view, opened and closed. It is also possible to abstain from the use of the categories, and display all property groups alphabetically instead.

The sorting method is set using this button. You can sort either by "category" or "alphabetically".

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

To edit a property, click in the value column. When standard values are used, the input of the new value is accomplished using a Combobox.

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

It is also possible, for almost all properties, to enter the value as a formula. To do this, click during the input of the value on the Combobox entry "Formula". For values that can be entered directly, click on the

16

button and a formula dialog will open. You will find a detailed description of this function in Chapter 2.8. Variables, Formulas and Expressions.

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

1.2.12. Using the Keyboard

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

Tool Window

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

Tool Window	Shortcut
Workspace	CTRL+1
Variable List	CTRL+2
Layers	CTRL+3
Objects	CTRL+4
Preview	CTRL+5
Properties	CTRL+6

Properties

You can also, if you so desire, work with key combinations within the property Lists.

Function	Shortcut
Change to next property or header	Cursor down
Change to previous property or header	Cursor up
Open category	+
Close category	-
Open all property groups of a category or sub-groups of a group	*
Edit property	Tab

Function	Shortcut
End editing	Shift + Tab
Open Combobox	Ctrl + F4 (in Edit mode)
ENTER	Depends on value: open content dialog, invert Yes/No, open Combobox

1.2.12.1. Drag & Drop

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

1.2.12.2. Online Help

The Designer has been equipped with an extensive online help that can be opened using **? > Contents** or **? > Context Sensitive**.

A Help button is also located in the bottom left corner of all dialog windows which opens the help for the appropriate dialog.

1.3. **Dialogs**

The dialogs of the List & Label Designer are designed with recurring functions, enabling you to quickly become familiar with the Designer and work effectively. The following list describes some buttons which will simplify your work in many dialogs:

	Add new contents to a dialog, for example a new line in a text object
	Edit the contents or properties of the selected object(s)
	Delete the selected object(s)
	Cut the selected object(s) to the clipboard
	Copy the selected object(s) to the clipboard
	Paste the object(s) from the clipboard to the current object

	Change the order of the selected object(s)
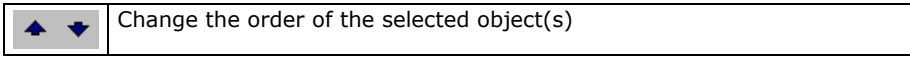
2. General Methods and Procedures

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

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

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

The individual elements of a project are called "**objects**". In this manner, a label project can, for example, consist of an object for the sender line, the address area, and the logo.

These objects contain the information required for printing such as the actual contents, fonts, alignments, word wrapping, colors, etc.

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

Texts
These objects can contain fixed text as well as the variable content of individual records. You can define the fonts, object orientation, and alignment of the contents.

Rectangles
These objects are rectangular frames for which layout options such as frame width, colors, filling pattern and shading can be defined.

Circles and Ellipses
Define Circles and Ellipses. Layout options such as frame width, colors and filling pattern can be defined.

Lines
With this object type you can define lines with layout options such as line width and color.

Picture Files
These objects allow you to integrate graphics into projects. These can be either fixed graphics (ex. a company logo) or variable graphics (i.e. different for each record).

Barcodes
These objects allow you to display fixed text or variable contents as barcodes. You can define characteristics such as the type of barcode, color, text, and orientation.

Tables (Lists)
These objects allow you to display fixed text and variable contents from various records. You can define layout characteristics such as appearance, fonts, column widths and alignments, word wraps, footers, etc.
Tables appear in list projects only.

Formatted Texts
With this object type you can change the format within a line.

Charts
With this object you can create diagrams.

HTML Texts
With this object you can display the contents of web sites and other HTML formatted text.

OLE Containers
Serves as a container for various OLE-Server-Documents. In this way, you can integrate documents created with Word, Excel, Visio or MapPoint into your project. See Chapter 5.13. Insert OLE-Server Documents for more details.

Form Templates
These graphic objects can only be inserted by using **Objects > Insert > Form Template**. They are placed, as templates, in the background of the workspace and are used for the exact placement of objects. These templates are helpful in the design of complex forms. The form templates are exceptional in that they are not printed later.

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

In order to edit an existing object you need to select it. Just click into the required object and the object will be selected. A selected object can be identified by its raised frame. When a new object is created, it is automatically "selected" and can be edited or changed immediately. The property dialog of an object can be activated with a double-click. Please be aware that the selection tool needs to be activated. See Chapter 1.2.2. Object Bar.

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

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

These steps will be described in more detail in the following chapters. New users can use this chapter as a guide. We will, naturally, go into most of List & Labels functions in this manual. For your definite project, you will probably only need a fraction of these functions.

2.1. **Projects**

2.1.1. Creating or Opening a Project

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

Create a new project

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

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

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

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

The Project Wizard

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

• page numbering
• page title (first page only or all pages)

- zebra pattern for the table
- create a summary
- select the data for the table object

Opening an Existing Project

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

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

A dialog will open in which you can select the file to be opened.
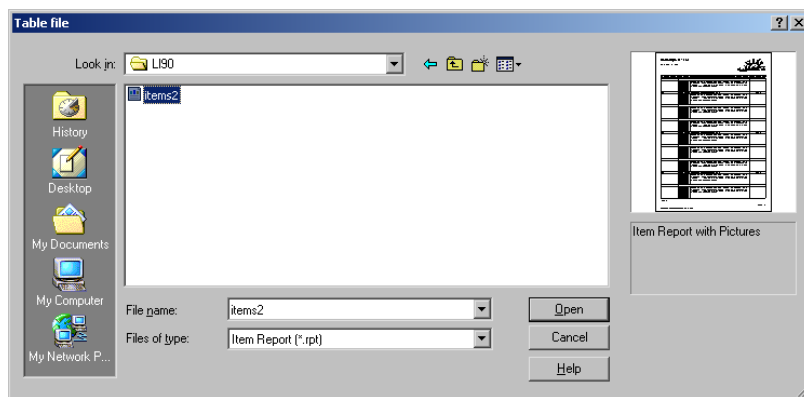
If a description has been given to a project (see **File** > **Save As**) the description will be displayed underneath the sketch field. In the "sketch" field you can see a sketch of each selected project, making it easier for you to find the project you are looking for. (The sketch is only shown if the corresponding option is set, see: **Options** > **Compatibility** > **Sketch in File Dialog**).
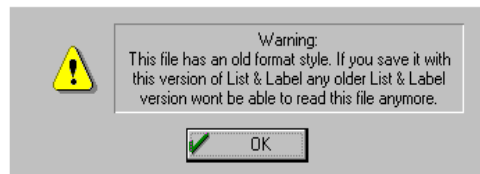
The Designer will issue a short warning if you select a project that was designed with an older version. Please note that projects designed with the new Designer will not be readable or editable with an older version. We advise you to make backups of your project files before editing them with the new Designer. Please check your layout after conversion carefully, as slight changes may occur.

2.1.2. Project Import

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

2.1.3. Project Types

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

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

2.2. **Defining Page Setup**

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

2.2.1. Page Setup for Labels

In the page setup for labels, you can define the paper size and orientation as well as the printer via the printer setup. You can either open one of the predefined label formats or define one of your own.

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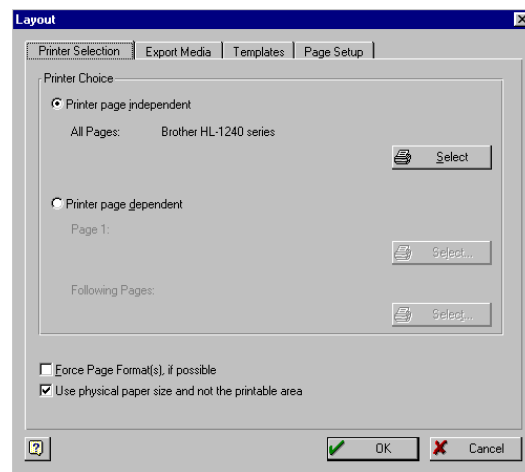
Click the "Select" button to define the paper size and orientation and select the printer for the project.

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

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

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

corporate letter-head paper for the first page and normal paper for the following pages.

Force Page Format(s), if possible

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

Use physical paper size and not only the printable area

If this option is set the whole physical page is available as workspace, including non-printable margins. Some label page layouts require this, as the first label starts right at the top left edge instead of at a specific distance. The unprintable margins are shown shaded in the full-page preview.

This means that objects can also be placed in the margins which will be cut off during the print. If objects are placed on page edges you still should take notice of the unprintable area.

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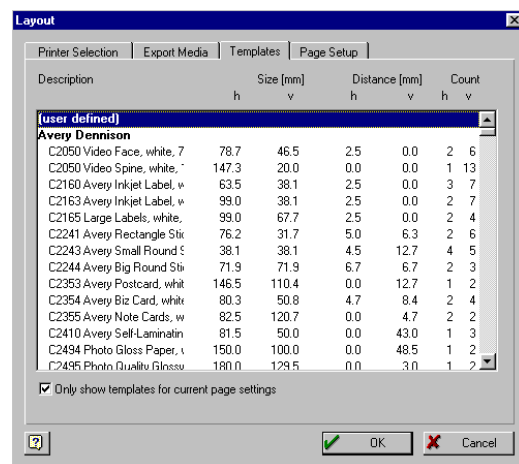
You can select from a range of pre-defined label formats in the "**Templates**" register in the page setup for labels. Now you won't have to figure out the size of a particular label and how many labels are on one page.

Defining Your Own Label Formats

It is also possible to define your own label layout if the one you require is not available. Use the following values:
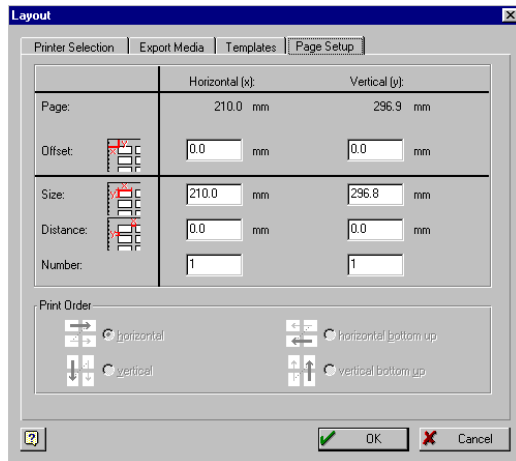
Setting the Offset

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

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

Setting the Size

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

Setting the Distance

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

Setting the Number

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

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

Defining Print Order

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

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

Defining Export Media

On this card you can specify which export target (Preview, HTML, RTF, PDF, XML, XLS, Picture, available options may vary) should be used as default for the project.

2.2.2. Page Setup for Lists

You can define the page setup for one of the printers you wish to use as well as the paper size and alignment.

You can also define whether the workspace should be displayed as a physical page or only with the areas which are actually printed by the printer selected.

Selecting Printer and Paper Format

The remaining set up is done just as with labels.

2.3. Default Values and Options

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

You can also open this options dialog via the menu item **Options**.

Each option in this dialog is explained by a tooltip.

2.3.1. Options for a Project

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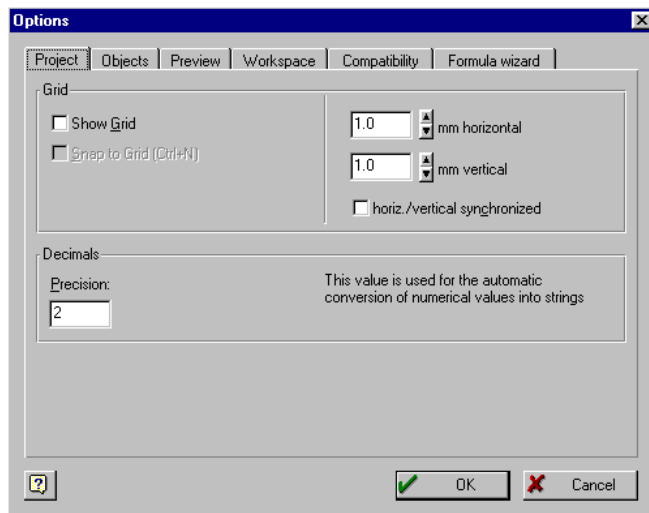
Defining Alignment Grid

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

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

Precision

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

2.3.2. Options for Objects

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

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

The default values are only valid for the project currently open.
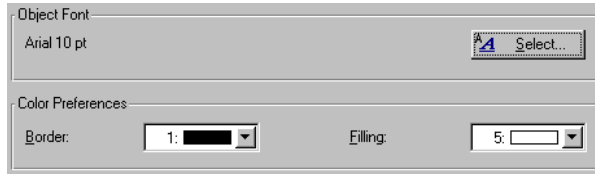
Object Font

In the "**Object Font**" group you can define the font for each newly inserted object, using the "**Select**" button.

These default values are only valid for objects which contain text, in this case text objects and list objects. The settings also affect all objects which were not changed manually to a non default style.

Object Colors

With the Comboboxes "**Border**" and "**Filling**" you can set different colors for different objects. These values are, however, only valid for the object types circle/ellipse, rectangle, and line.
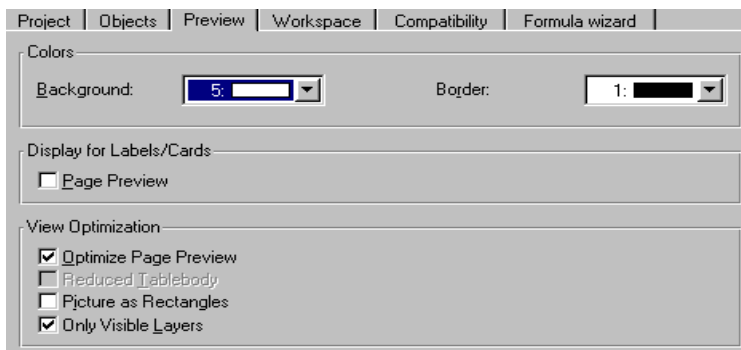
2.3.3. Options for the Preview

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

Setting Colors for the Preview

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

View of Label/Card Projects

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

Optimized View

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

2.3.4. Options for the Workspace

Setting Rulers

Displaying Objects

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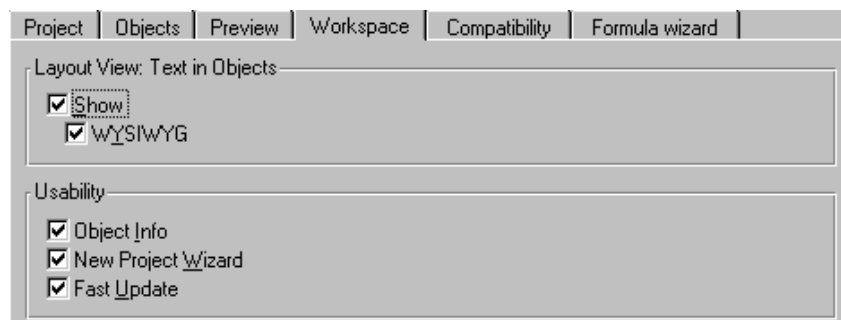
contained in objects should be displayed or not on the workspace. The "**WYSIWYG**"-option displays the selected fonts and formats the way they are printed.

Settings for Usability

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

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

2.3.5. Compatibility Options

Here you can set various options which concern or are involved with other software programs or hardware components.

Clipboard Format is Text

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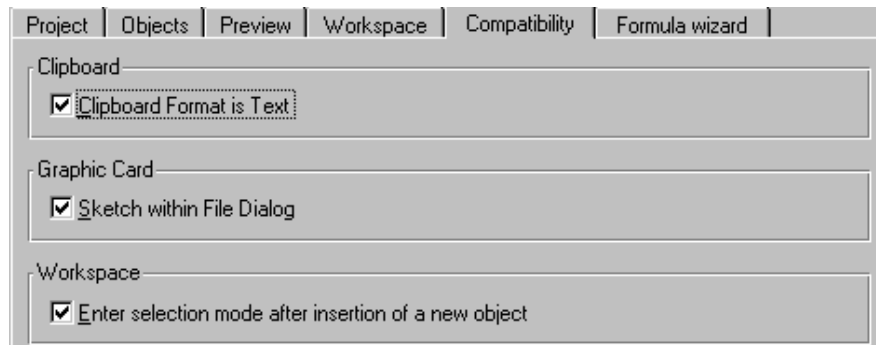
pasting Designer objects into other applications.

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

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

Sketch within File Dialog

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

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

Compatibility with former Versions

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

2.3.6. Options for the Formula Wizard
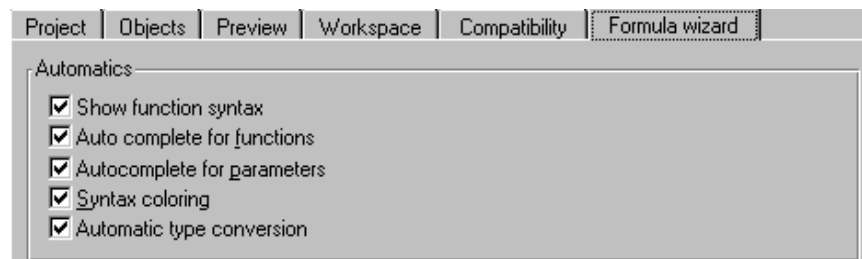
The formula wizard offers many enhancements for creating formulas or functions. Detailed information about working with formulas can be found in Chapter 2.8. Variables, Formulas and Expressions. Here you can disable these enhancements.

- Show Function Syntax: a tooltip describing the function and parameters appears.
- Autocomplete for Functions: after entering 2 characters functions that fit will be shown and can be selected.
- Autocomplete for Parameters: after entering 2 characters parameters that fit will be shown and can be selected.
- Syntax Coloring: functions, parameters and operators are displayed in different colors.
- Automatic Type Conversion: variables and fields will be surrounded by conversion functions to fit the context of the function.

2.4. Inserting Objects

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

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

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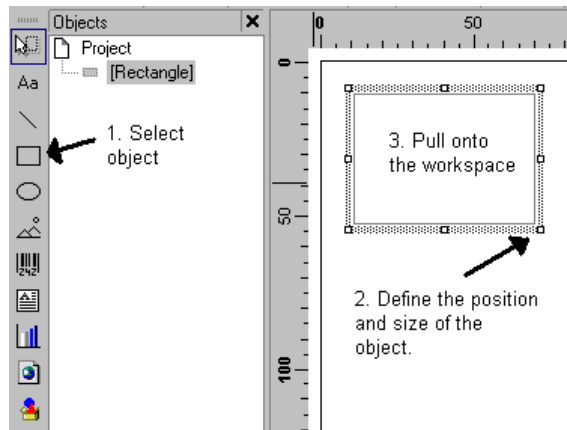
2. Place the crosshair on the point at which a corner of the object should be placed. It is best to use the left upper corner of the planned object. Depress the left mouse button and pull – while keeping the mouse button depressed – to the opposite corner of the planned object. If you started in the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

3. Release the mouse button when the object (dashed frame) has the desired size.

2.5. Editing Objects

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

- change the size of the object
- reposition the object
- define the exact position of the object on the workspace via the **position dialog** in 1/10 millimeter increments
- edit the contents (properties) of the object. These contents are different according to the type of object selected (text, picture, table, etc.). You can find a detailed description of the properties of objects in the subchapters of Chapter 5. Menu Objects
- define appearance conditions for the object. With an appearance condition you define the condition(s) that must be met for the object to be printed. In this way you can allow a logo to appear on the first page of a project only and not on the following pages
- name the object. Using the name you can easily find the object you are looking for
- assign the object to a (display)-layer or copy it to a layer. Multipage and/or complex projects become clearer if objects which belong together are put on a mutual layer. So, in a multipage project all objects which belong to one page can be put on a common layer. You only need to switch the visibility of this layer when you want to edit it

- copy the object. If you want to place several, similar objects with the same distance on the workspace then you can use the function **Create Multiple Copies**.
- If you have selected more than one object you can combine the selected objects into a **group** (grouping), align them or adapt their size.

These editing possibilities will now be described individually in detail.

2.5.1. Selecting Objects

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

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

Selecting Multiple Objects

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

2.5.2. Moving Objects

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

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

Changing Size

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

- Release the mouse button when the object has reached the required size.

Repositioning an Object

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

Changing Size and Position via Dialog
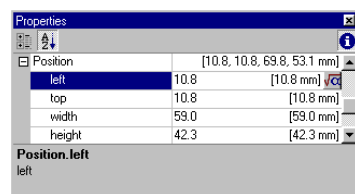
You can change the size and/or position of an object in the Properties List. Please note that changing an entry will cause the automatic recalculation of all other related values.
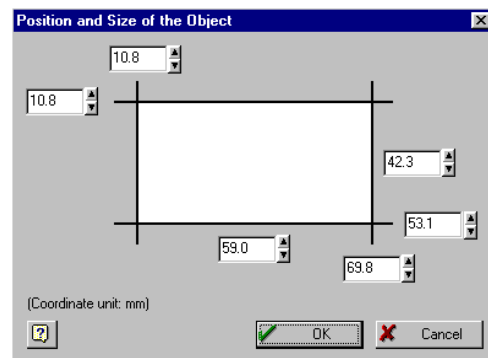
If you double-click on the subcategory "**Position**" in the Properties List a position and size dialog opens. Using this dialog, the defining of the size and position of an object using the keyboard is simplified.

Repositioning and Changing Size via the Keyboard

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

- Select the object you wish to reposition.
- Use the CURSOR-KEYS to move the object in the direction you require. Press the key once and the object moves 1/10 millimeter, if the SHIFT-KEY is also held, the object is moved by 1 millimeter.
- Use the CTRL and CURSOR keys for fine adaptation of the object's size

- Use the SHIFT, CTRL and CURSOR keys for a quick adaptation of the object's size.

2.5.3. Grouping Objects

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

To combine two or more objects use the following procedure:

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

2.6. Working with Layers

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

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

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

2.6.1. Defining Layers

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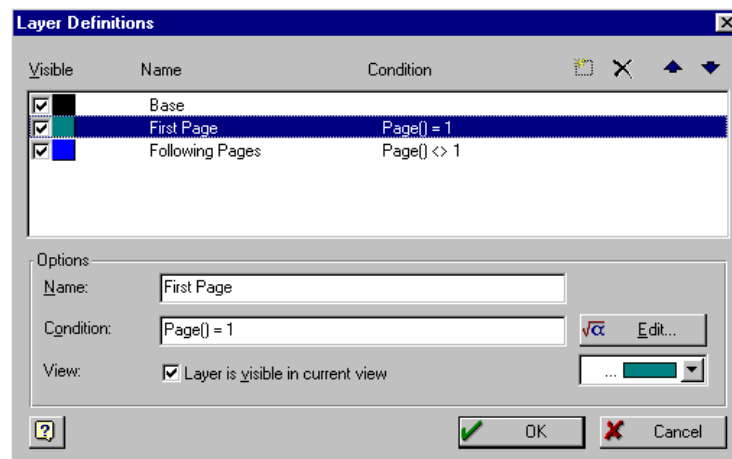
Inserting, Deleting and Editing Layers

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

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

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

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

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

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

Appearance Conditions for Layers

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

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

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

Typical appearance conditions for layers are those that allow objects of a layer to appear only on certain pages.

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

Additionally, logical expressions can be used as appearance conditions.

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

2.6.2. Assigning Layers

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

Select the objects which you wish to assign to a particular layer, and then select the command **Assign to Layer** from the **Objects** menu or the context menu. A list with the defined layers will appear. Select the desired layer.
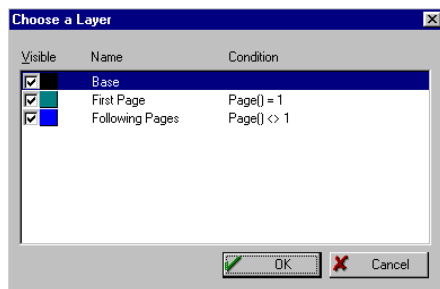
Double-click on the required layer to assign the selected objects to it. Alternatively, you can also click on the required layer and then confirm the dialog with "OK".

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

2.6.3. Copying into Layers

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

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

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

Switching Layers On/Off

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

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

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

2.7. Object Properties
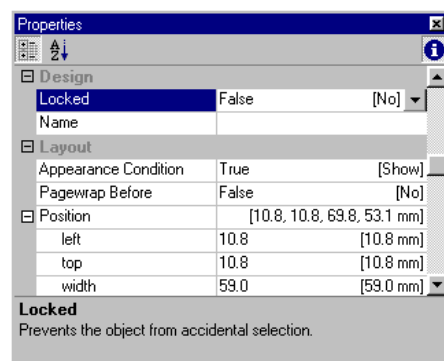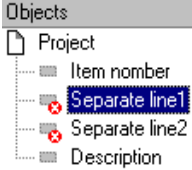
Most properties can be defined using the Property list or additional dialogs. The properties for every object are different, but there are a series of characteristics that all object have in common, for example size, position, name and appearance conditions. The properties for individual objects are described in detail in the relevant subchapter of Chapter 5. Menu Objects.

When predefined values are

available for a property, a button ▾ will be available in the second column of the list. This button opens the list of predefined values for the property.

2.7.1. Locked

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

Tip: A locked object can, as always, be selected in the Object List for editing.

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

Properties

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

2.7.2. Name

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

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

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

Properties

Property	Description	Value	Description
Name	Object Name	Name	

If you have activated the option **Options > Workspace > Object Info**, the object name will be displayed in the tool tip.

2.7.3. Appearance Conditions
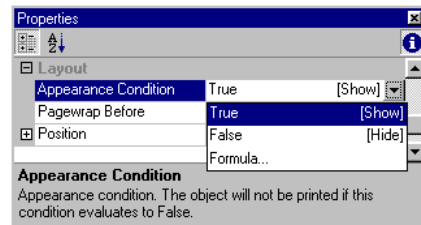
An appearance condition can be set, for every object, that defines under which conditions the object will be printed . These appearance conditions are listed in the Properties List under the category "**Layout**". These are object specific filter conditions that are further described in Chapter 2.8.2. Using Filters.

Properties

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

You will find guidance for the definition of these conditions in Chapter 2.8. Variables, Formulas and Expressions

2.7.4. Page wrap before Object Output

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

Properties

Property	Description	Value	Description
Wrap before	If the condition "True" is returned, a page wrap will be initiated before printing of the object.	True False Formula	Break No break Formula dialog

2.7.5.Position

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

Properties

Property	Description	Value	Description
Position	Position and size of the object, all data is entered in the unit used on the workspace	...	Position dialog
Left	Horizontal offset of the upper left corner of the object with the upper left corner of the workspace	Number Formul	Formula dialog
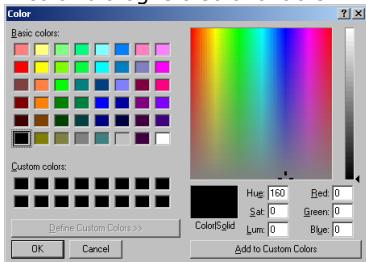
Property	Description	Value	Description
		a	
Top	Vertical offset of the upper left corner of the object with the upper left corner of the workspace	Number Formula	Formula dialog
Width	Object width	Number Formula	Formula dialog
Height	Object height	Number Formula	Formula dialog

2.7.6. Colors

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

Properties

Property	Description	Value	Description
Color	The color can be selected from either the list of predefined colors or by using a formula . The color consists of a red, green and blue portion and is selected using the RGB (Red Green Blue) function. Each color portion can assume a value between 0 and 255. 0 denotes no portion of this color, 255 denotes full color saturation. The three portions are mixed additively, with RGB(0, 0, 0) denoting black, RGB (255,255,255) white. A color dialog is also available. 	... Predefined color list Formula	Color dialog Choice of predefined colors Formula dialog

2.7.7. Contents of Objects

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

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

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

You will find further information about the Contents dialogs for individual objects in Chapter 5. Menu Objects.

2.8. Variables, Formulas and Expressions

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

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

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

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

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

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

2.8.1. The Elements of an Expression

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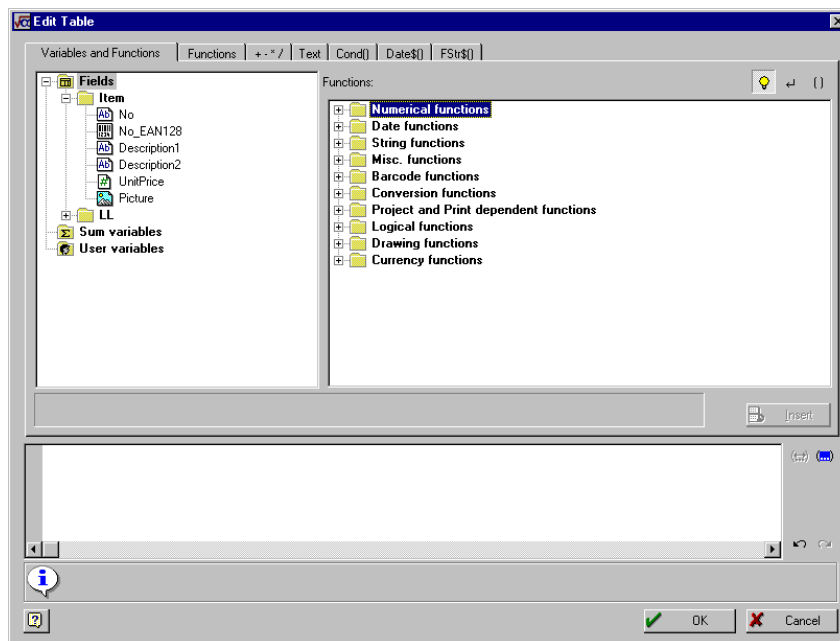
Input Enhancements of the Formula Wizard

The formula wizard supports you with various input enhancements, that can be (de)activated under **Options > Formula Wizard**:
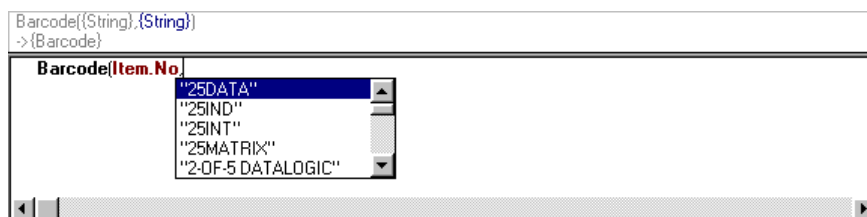
- Show Function Syntax: if this option is selected, a tool tip appears in the formula assistant that describes the selected function, lists the necessary parameters and displays the result types.
- Autocomplete for Functions: if this option is activated, after entering 2 characters, all available functions that begin with those 2

characters will be shown and can be selected. This expedites the entry of the function.

- Autocomplete for Parameters: the assistant recommends values based upon the available variables and fields.
- Syntax Coloring: functions, parameters and operators are displayed in different colors. This will improve the transparency in complex formulas.
- Automatic Type Conversion: if this option is activated, variables and fields will be will be automatically converted at the time of entry of the expression so the type of data conforms to expectations.

Expression Modes

Please note that there are two possible modes for expressions:

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

On the other hand there is the extended mode, where fixed text may be stated as is without quotation marks. Variables are framed by "<" and ">" and functions are framed by chevrons "«" and "»". The chevrons can

be inserted by clicking the button ⟪⟫. The combining of variables with the "+" operator is not necessary. This mode is easier to use.

Register Cards

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

Card	contains the elements
Variables and Functions	the variables and functions available for this object type
Function	a list of the functions available
+ -*/	a list of the available link operators
Text	a dialog for the entry of fixed text
Cond()	a special dialog for the definition of IF-THEN-ELSE conditions
Date$()	a list of the available date formats
Fstr$()	a list of the available number formats
Tab	the setting possibilities for tabs (only available in text objects)

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

There are special rules for the syntax and linking of various individual elements of an expression (variables, text, functions, etc.). For this reason you should always insert the various elements into the expression

field via the relevant card of this dialog. In order to insert variables you should use the "**Variables and Functions**" card, to enter text the "**Text**" card, etc.

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

The Expression Field

«Cond(AVAILABLE,"Article is available")»

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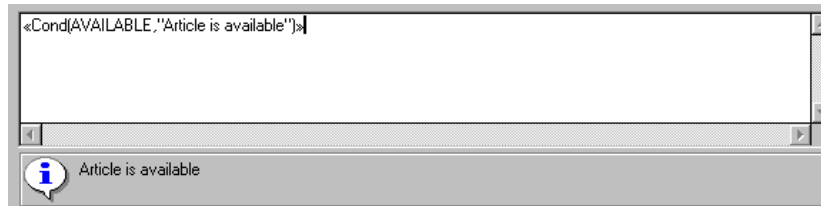
The expression in the expression field is constantly checked during its "creation" for correct syntax. Any syntax errors are displayed in the info field beneath the input field, along with a hint as to the cause of the error. As long as the expression is incomplete the check routinely displays at least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

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

The three buttons next to the input field are used to

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

Inserting Variables

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

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

Double-click (or use drag & drop) on the variable you'd like to transfer to the editing line.
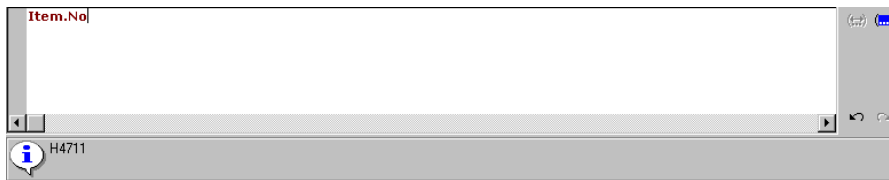
The required variable is then transferred to the editing line with the correct syntax.

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

FIRSTNAME + " " + LASTNAME

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

Insert fixed text

Another very important element in expressions is fixed text, for example as prefix for variables like

"Phone Number " + PHONE

which will print :

Phone Number 1-555-123-456

The card "**Text**" offers an easy way to insert text automatically, set tabs and set line breaks.
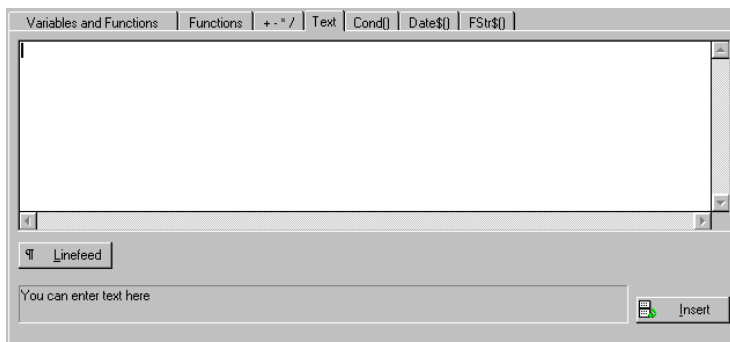
Enter the text and click on the "**Insert**" button to insert the text into the expression below. The text will be inserted automatically, depending upon the requirements, either in quotation marks (for function expressions) or without (for text expressions). For this reason you should always enter text using the "**Text**" card.
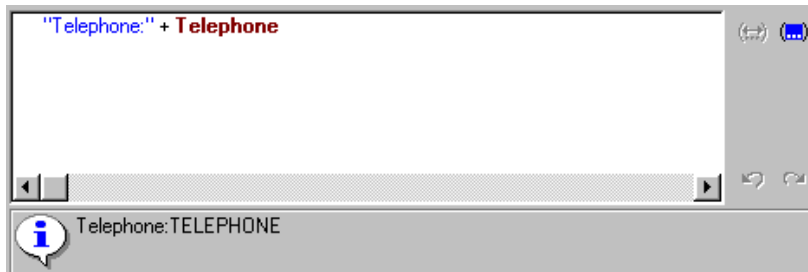
In this example, the fixed text "Telephone" was entered using the "**Text**" card, and then the variable TELEPHONE using the "**Variables and Functions**" card. The text "Telephone" will be printed first followed by the appropriate telephone number from the database.

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

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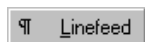
 To insert a linefeed, click the button "**Linefeed**". The linefeed is represented by a special character.

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

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

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

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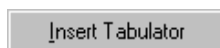
 Tabs are only available in text objects, thus the button will not appear in fields of table columns.

Only one tab can be inserted for each line. Its position and alignment can be adjusted using the card "Tab". For more information please refer to Chapter 5.3. Text Objects.

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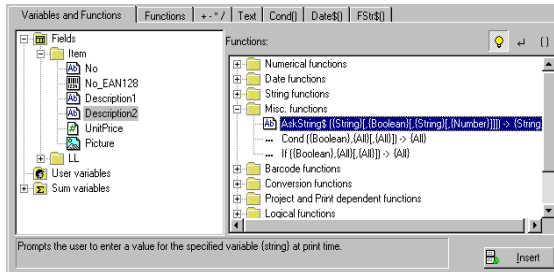
The built-in functions offer flexibility in defining expressions. Functions allow you to change the representation of fixed or variable text or numbers, recalculate and modify values and to set a display format for these.

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

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

- function group (e.g. "Numerical Functions", ...)
- type of return value (e.g. "Numerical Value",...) or
- type of the first parameter (e.g. "String",...)
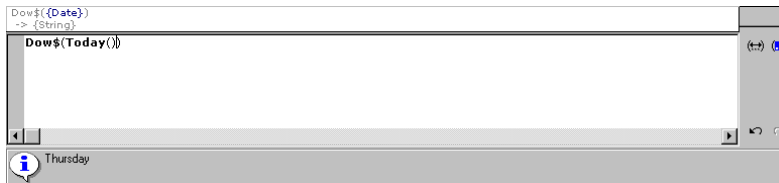
In this example, the user clicked the "**Insert**" button for the "dow$()" function which returns the day of the week of the date that is passed in the parameter of the function, ex. "Thursday".

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

You can also use functions for some parameters. For example:

dow$(Today()))

which will print the current day.

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

Insert a logical condition

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

Let us assume that your data base contains both house and post office box numbers, and that you would like to print address labels using this data. Naturally, only the street address or the post office box number should be printed, but not both. You decide to use primarily the post office box number and only to use the street address when the post office box number is not available. With the help of a IF-THEN-ELSE condition, this will be no problem
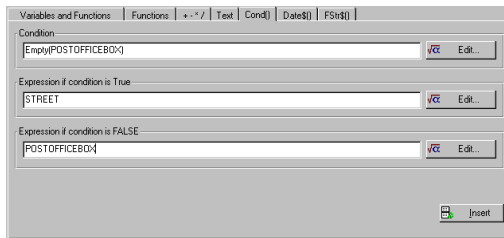
Enter the condition to be evaluated, or a Boolean variable, into the condition field. Enter the "true" part into the second line, and the "false" part into the third. Note that the "**Edit**" button starts an additional assistant to help you edit the appropriate part. In our example, we used the function Empty() to evaluate if the POBOX field is empty. When you press "**Insert**", the parts will be wrapped into the "Cond(...)" function and inserted into the expression line below.

In the field "True", enter the expression to be used if the entered condition is TRUE. If nothing should be done, enter two consecutive quotation marks ("").

In our example, the condition is true if the POSTOFFICEBOX field is empty, and in this case the STREET should be printed.

In the field "FALSE", enter the expression to be used if the entered condition is FALSE. If nothing should be done, enter two consecutive quotation marks ("").

In our example, the condition is false if the POBOX field is not empty, and in this case the POBOX should be printed.

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

Formatting Dates

Using the dialog with the card "Date$()" you can get assistance for date formatting. You can have full-text formatting for day and month values, 2- or 4-letter representation of the year and so on.
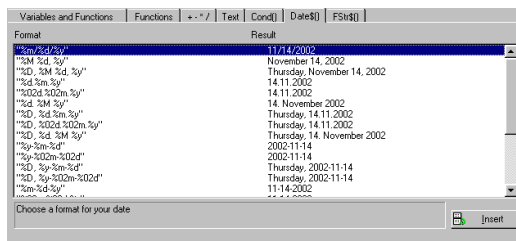
To the left is the format string, to the right the result with a date example:

As default, the function Today() is selected as date parameter which will display the current date. Of course you can replace it with any date value you like:

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

it would display the date one week from today.
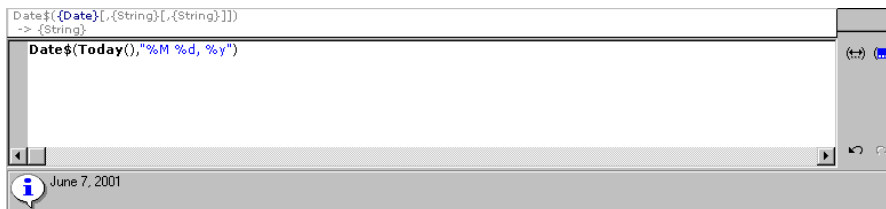
In the above example, the date function "Today()" has been so formatted so that the result is displayed with the month written out, the date, and the year as a four digit number. More detailed information on the use of the Date$() function can be found in the appendix.

Formatting Numbers
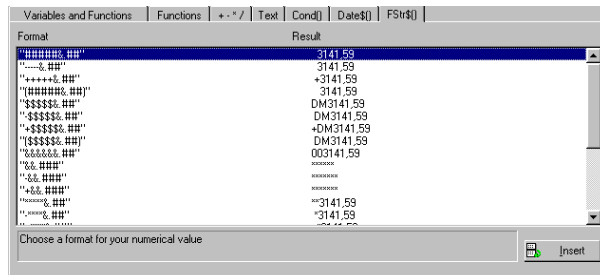
Using the card "FStr$()" you can define the format in which a number shall be presented. In this way number of digits, decimals, leading signs etc. can be defined.
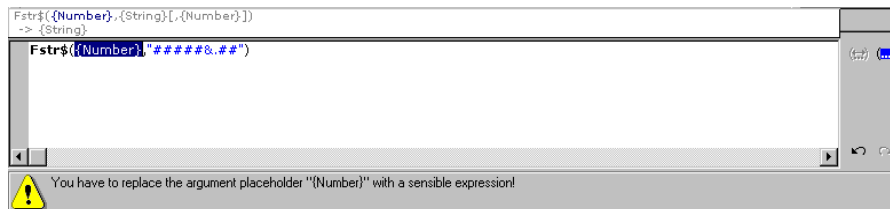
In this example, the price will be displayed with 6 digits including 2 decimals. If the number just before the decimal point is 0, only one '0' will be displayed.

Additional information about the function "FStr$()" can be found in Chapter 7.1.3. Examples for the Use of Functions.

Operators

The card "**+ - * /**" displays a list of all available operators. These can be used to make calculations, combine variables and/or fixed text, as well as to perform logical comparisons.
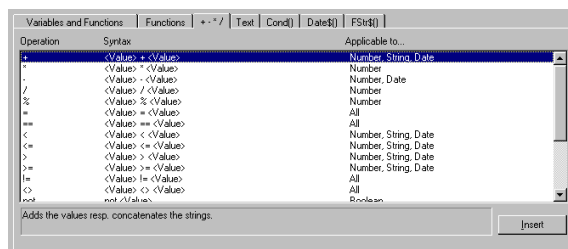
The available operators can be categorized, in order of increasing priority, into logical operators (AND, OR, NOT), arithmetic operators (+ , -, *, /, %) and relations (<, < = , = , >, > = , <>, ! =).

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

There are three columns on the dialog page. The first describes the operator, the second is syntax and the third the type(s) of value(s) it can be applied to.
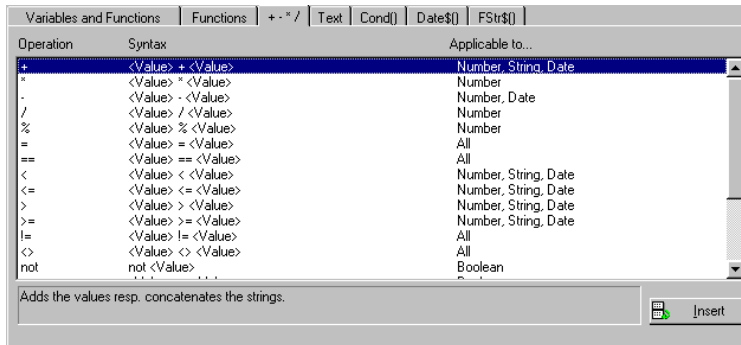
The operator "+" has particular importance, because it can not only be used for addition (Value types "Number" and "Date"), but also for the concatenation of text (Value type "String") or fixed text.

Working with Sum Variables

Sum variables can be used to build sums/totals over the records, for example the sum of the column PRICE in a table.

It is possible to optionally add all records on a particular page (page totals) or for the whole project (totals).

Defining Sum Variables

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

@SumXX = @SumXX + <numeric expression>

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

For each record which has been printed, List & Label calculates the value from "<numeric expression>" and adds it to the "@SumXX".
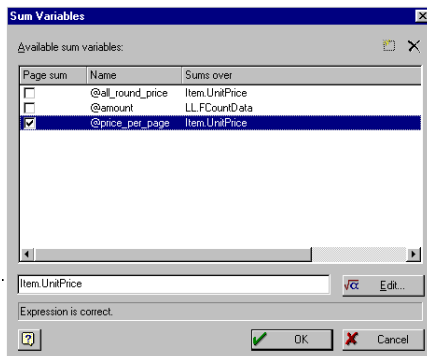
To use sum variables of this kind ("@SumXX") in projects they must first of all be given the required numeric expressions. This is done with the command **Project > Sum Variables**.

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

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

@SumXX = @SumXX + <PRICE>

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

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

Defining a Counter

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

In the most simple case, a counter is increased for each record.

@SumXX = @SumXX + 1

Page Totals and Other Totals

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

Using Sum Variables

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

To do this, open the Contents dialog of the appropriate object and then open the dialog for the definition of expressions. (See Chapter 2.8. Variables, Formulas and Expressions). In the card Variables and Functions, you will also find the Sum Variables @SumXX listed. You can use these variables in your expression just like any other variables.

2.8.2. Using Filters

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

Filter conditions are logical expressions and the result determines whether the object or record will be printed. If the result of the logical expression is TRUE, the record is used (or the object is printed). If the

result of the logical expression is FALSE, the record or object will not be printed.

2.8.3. Record Filters

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

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

Project Specific Filters for Records

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

To assign a project specific filter, use the command **Project > Filter**, which will open a dialog for the definition of filter conditions. This is similar to the dialog described in Chapter 2.8. Variables, Formulas and Expressions.
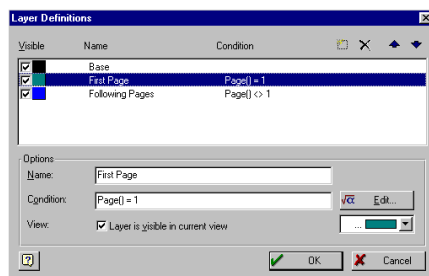
Layer Specific for Objects

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

Select the level, in the Layer Window, to which you wish to assign layer specific appearance conditions. Enter the desired condition directly into the field "**Condition**", or open the dialog for the definition of layer conditions (see Chapter 2.8. Variables, Formulas and Expressions).

Object Specific for Objects

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

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

2.9. Saving projects

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

2.9.1. Save

To save a project select **File > Save**.

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

2.9.2. Save As

To save an existing project under a different name or path, select **File > Save As**.

A dialog will open in which you can enter a name for the project, the device and the directory in which the project will be saved.

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

2.10. Printing Projects

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

2.10.1. Print Sample from the Designer

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

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

Printing layers

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

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

Print Sample with Frames

By selecting the command **File > Print Sample > Print Sample with Frames**, your labels will be printed on plain paper with a frame, which represents the borders of the labels.

These frames do not appear when printing real data.

By using these frames, you can determine if your labels have been printed in the correct size and position.

Print Sample without Frames

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

Print Sample in List Projects

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

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

Print Sample 1st Page

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

Print Sample Following Page

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

Print Last Page

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

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

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

Please note that the condition "Lastpage()" is only reasonable for use with linked objects. Detailed information about linking of objects can be found in Chapter 5.2. Insert.

2.10.2. Printing to Real Data Preview

In addition to the Sample Print using sample data, List & Label offers a real data preview. But in contrast to the sample print, this can't be used in the List & Label Designer, but in the application. Refer to your application's manual for hints on how to open the preview print.

All print jobs can be viewed prior to printing in a preview window on the monitor. In this way, you can review the layout of your project without wasting paper. The output to the monitor takes place as it will during the actual printing. After reviewing the layout, you can start printing directly from the preview without again having to use the print command.
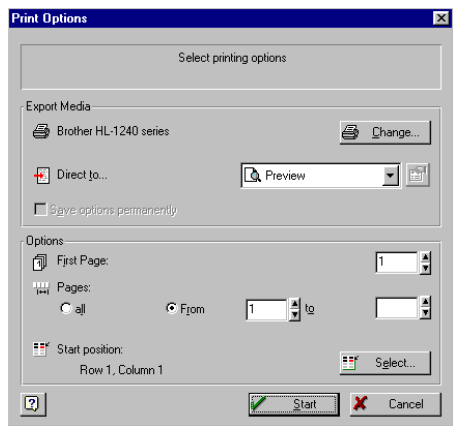
The dialog shown on the left will be presented which allows you to select the destination printer and - in case of label projects - the starting label.

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

In the category "**Options**", you can enter the number of copies, the starting page number and the page range to be printed.
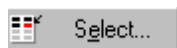
For label projects, you can additionally select the position of the first label on the first page. This is useful when printing on pages where labels have already been used.
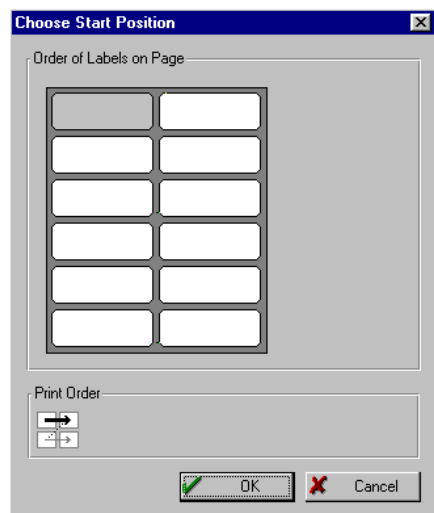
The dialog shows the layout of the label page. Click on the label you want to use first during printout, taking into account the print order noted below. You can print not only from the upper left to the lower right, but you can reverse this order or print by columns. Printing will start from the label that you chose.
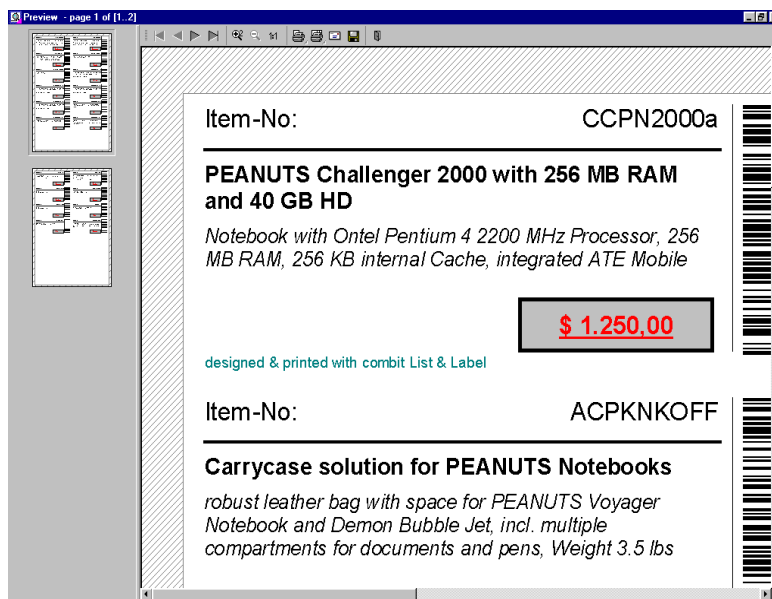
The entire print data can be viewed and printed in a preview window. The non-printable margins will be shaded. On the left side of the Preview

window, you will find the individual pages of your project in reduced size as "thumbnails". This allows a very fast, direct navigation through the document. When you click on a thumbnail, the appropriate page will be displayed in the preview.

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

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

The arrow buttons allow you to change to the next/previous page or jump to the beginning/end of the document.

The magnifiers, as well as the 1:1 tool, represent various zoom levels that can be used to see fine details in the document. Alternatively, you can use the mouse to select an area to zoom by drawing an rectangle.

Using the button "**Print Current Page**", you can individually print the currently displayed page from the preview. If you right click on this button, a Printer Assignment dialog window is opened. Here you can set the printer to be used, the page range and the number of copies to be printed.

Using the button "**Print All Pages**" will send all pages directly from the preview window to the printer. The file will be printed exactly as shown in the preview. The Printer Assignment dialog can also be opened using the right mouse button.

Use the button "**Send To**…" to send the current preview file (*.LL) through your MAPI-Mail-Client to an email recipient. In order to view the preview, the recipient must use the List & Label Viewer, which is often included with the application. The List & Label Viewer can also be usedto print the preview file.

Using the button "**Save As**…" you can save the current preview file with the *.LL file extension. Alternatively, you can save the individual pages of the file in Metafile (.EMF) format.

Using the "**Exit Preview**" button, you will leave the preview window and return to your application. The displayed print preview will not be printed, but discarded. Any changes will be lost unless previously saved with "**Save As**…".

2.11. Exiting the Designer

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

3.Menu Edit

In the menu "**Edit**" you will find the normal commands, i.e. Undo, Delete, Cut, etc., known from other Windows based programs.

4. Menu Project

4.1. Page Setup

When you select the command **Project > Page Setup**, a dialog opens in which the configurations possibilities for your type of project – lists or labels – are presented. Configuration of the page layout, printer selection and paper size and format is available.

4.2. Filter

You can define the appearance conditions for the lines of a text or table object using the command **Project > Filter**. These conditions determine whether a record will appear in your project or not.

4.3. Sum Variables

Sum Variables are defined using the command **Project > Sum Variables**. Sum variables can be used to build sums/totals over the records. Sum variables are identified with @ <NAME> (i.e. @Summe01).

4.4. User Variables

Formulas that you often need in the same form can be saved in so called user variables. User variables are identified with @<Name> (e.g. @User01).

4.5. Fax Variables

You can send List & Label documents as a fax using the Windows 2000 / XP fax printer. If a fax capable modem is installed on your computer, the appropriate driver will be set up automatically by Windows 2000 / XP. Additional information is required to properly address and send the fax, this means that at least the fax number must be contained in the document.

To define the fax variables use the menu command **Project > Fax Variables** and select the desired option.

4.6. Layer Definition

The menu command **Project > Layer Definitions** opens the dialog for the definition of appearance layers. This is useful for complex or multipage projects.

4.7. Options

The command **Project > Options** opens the options dialog. Options are described earlier in this manual.

5. Menu Objects

This menu contains various functions for the editing of objects.

5.1. Select

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

5.1.1. Selection Mode

A detailed description of how you can select object in the Object Mode can be found in Chapter 2.5.1. Selecting Objects.

5.1.2. Select All

Use the command **Objects > Select > Select All** or the shortcut CTRL+A to select all objects in the workspace.

5.1.3. Invert Selection

Use the command **Objects > Select> Invert Selection** or the shortcut CTRL+Y to invert the selection. All selected objects become unselected and all unselected objects become selected. This is valid only for objects on visible layers.

5.1.4. Next Object

The order in which the objects were created is important for the command **Objects > Select > Next Object**. This command, or the PLUS key on the numeric keypad, selects the object that is next in the object list (that is, the object created later) than the currently selected one.

5.1.5. Previous Object

The opposite of the command above.

5.2. Insert

Using the command **Objects > Insert**, a sub-menu opens in which you can select the type of object to be inserted into your project. A general description of the types of available objects was provided in Chapter 2.7. Object Properties.

A description of the properties that all objects have in common can be found in Chapter 2.7. Object Properties.

5.3. Text Objects

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

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

5.3.1. Properties

Property	Description	Value	Description
(Contents)	Opens the Contents dialog	🖼	Opens dialog
Bottom Aligned	Alignment at the bottom of and within the object frame. If this option is activated the object's text will be aligned on the lower edge, if not, on the upper edge. Requirement is that the paragraphs are not larger than the object. If so, the text will be clipped. This option is useful when text of an unknown length is to be placed at the bottom of a page.	True False Formula	Bottom Aligned Top Aligned Opens Formula dialog
Appearance Conditions	See Chapter 2.7.3. Appearance Conditions		
Rotation	Rotates the object counter-clockwise. Please note, that only TrueType fonts can be rotated.	0 1 2 3 Formula	0° 90° 180° 270° Formula dialog
Locked	See Chapter 2.7.1. Locked		
Background	This option allows text objects to be assigned a background color. When the value is "True", a background color can be selected using the "Color" option.	True False Formula	With background color without background color Formula dialog

Property	Description	Value	Description
Color	Background color See Chapter 2.7.6. Colors		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Page wrap	Defines if the object can initiate a page wrap. If the property is activated contents that exceed the size of the object will automatically appear on the next page. This is useful for text objects that occupy multiple pages. With labels, the next label will not be started until all objects with this option have been printed on the current label. It is possible that this option cannot be used if page wrap is not supported by your application.	True False Formula	Yes No Formula dialog
Wrap before	See Chapter 2.7.4. Page wrap before Object Output		

5.3.2. Text Contents

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

The individual paragraphs can be edited and contents entered in the Paragraph Properties dialog.
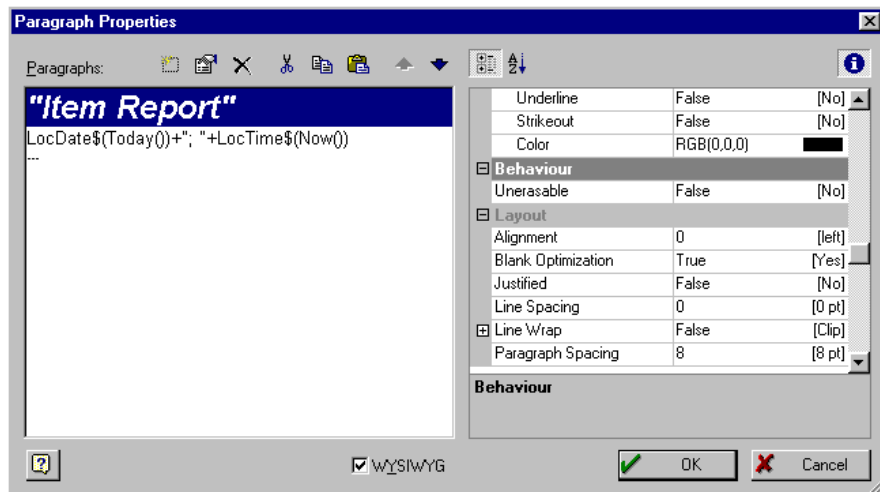
5.3.3. Editing Paragraphs

Text objects are paragraph oriented, this means that each paragraph can be individually edited and formatted. To edit multiple paragraphs at the same time, they must be selected beforehand.

With these buttons you can:

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

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

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

WYSIWYG

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

Edit Paragraph

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

This opens the Formula Editor that has an additional card.

5.3.4. The Card Tab

Tabulators are formatted, after you have set them in the card "**Text**" in the general definitions dialog, by using the card "**Tab**". You can set both the alignment and position of the tabulator.

Tabulator Alignment
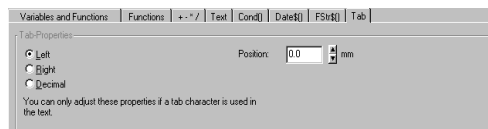
A tabulator causes the preceding text to run, to a maximum, to the tab stop. If the option "**Line Wrap**" in the Paragraph Properties is

selected, the text will be wrapped into the next line, if necessary. Otherwise, the text will be cut.

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

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

Tabulator Position

The tabulator's position within the selected object is given in millimeters. A positive value signifies that the position is calculated from the left edge of the text object. A negative value signifies that the position is calculated from the right edge of the text object.

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

Accepting an Edited Line

After you have defined the expression for the line, you can accept the line into your text object with the button "OK". You then automatically return to the dialog "Text Properties".

5.3.5. Paragraph Properties

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

Property	Description	Value	Description
Paragraph spacing	Space between paragraphs ("paragraph end spacing") The spacing is entered in points: To achieve a 1,5 line spacing with a 10 point font, enter 5 points. Negative values are allowed.	Number Formula	Formula Dialog
Alignment	Here you can, exactly as in word processing, define the alignment of individual lines of text.	0 1 2 Formula	left centered right Formula dialog
Block	Block text is both left and right justified. This property is only functional when the selected line is wrapped. This means that the line must be longer than the available	True False Formula	Yes No Formula dialog

66

Property	Description	Value	Description
	space, consist of multiple words and the option "**Line Wrap**" selected. The last line will be displayed in accordance with the property "**Alignment**".		
Blank-Optimization	Blank-optimization can remove undesired spaces (leading, enclosed and remaining). Imagine that you wish to print a label project using the following variables which are to be separated with a space: <TITLE> <FIRSTNAME> <LASTNAME> If there is a record in which no title exists, this variable would remain empty but the following space would be printed. First name and Lastname appear with a leading space. If, for example, the variable "Firstname" was empty, there would be two spaces between "Title" and "Lastname" (enclosed spaces). If all three variables were empty, both spaces would remain (remaining spaces). This line is then not empty and would not be automatically ignored. In these cases, the option "Space-Optimization" assists in the automatic removal of leading, enclosed and remaining spaces. Multiple enclosed spaces will be automatically reduced to one space.	True False Formula	Yes No Formula dialog
Font	If default is set, the default font will be used. The Font Definition dialog can be opened using the font button. All properties in this properties group can be set in one dialog.		Font dialog
Default	Default font will be used instead of the set values.	True False Formula	Default font No Formula dialog
Name	The name of the font can be selected.	Fonts list	
Character Set	Defines country interpretation of the character set.	Number	Character set
Size	Font size in points. A list of available sizes for the selected font will be shown.	Number Formula	Predefined size Formula-Dialog
Font Width	Font width. 0 signifies standard width, otherwise the average character width will be displayed.	Number Formula	Width Formula-Dialog
Bold	Selects/deselects the text property "**Bold**"	True False Formula	Yes No Formula dialog

Property	Description	Value	Description
Italic	Selects/deselects the text property "**Italic**"	True False Formula	Yes No Formula dialog
Underlined	Selects/deselects the text property "**Underlined**"	True False Formula	Yes No Formula dialog
Strike Out	Selects/deselects the text property "**Strike Out**"	True False Formula	Yes No Formula dialog
Color	Font color See Chapter 2.7.6. Colors		
Uneraseable	Lines that remain completely empty will be ignored. The following lines will be moved up. This is logical, but at times can be undesirable. With the option "**Uneraseable**" the empty line will remain if it would be empty after use of the variable.	True False Formula	Yes No Formula dialog
Line Spacing	Line spacing defines the space between individual lines of a paragraph. The spacing is entered in points: To achieve a spacing of 1.5 lines with a 10 point font, enter the value 5. Negative values are also allowed. This should be accomplished using **Options > Objects > Object Font.**	Number Formula	Formula dialog
Line Wrap	Allows text that extends beyond the right edge of the object to be wrapped into a new line. If this option is not selected, the text will be cut. (If the text consists of a single long word, the text will be cut, in this case the option setting is irrelevant).	True False Formula	Wrap Cut Formula dialog
Keep Together	With this property you define whether the paragraph will be kept together during a page wrap.	True False Formula	Wrap Keep together

5.4. Inserting Lines

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

5.4.1. Properties

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

Property	Description	Value	Description
			dialog
Appearance Conditions	See Chapter 2.7.3. Appearance Conditions		
Color	Line Color See Chapter 2.7.6. Colors		
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.5. Inserting Rectangles

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

5.5.1.Properties

Property	Description	Value	Description
Appearance Conditions	See Chapter 2.7.3. Appearance Conditions		
Filled	Define if the rectangle object should be empty or if it should be filled with a pattern and/or color.	True False Formula	Yes No Formula dialog
Color	Filling color See Chapter 2.7.6. Colors		
Pattern	Select one of a large number of predefined patterns. A number represents a pattern. This property will only be evaluated if "Filled" is set to "TRUE".	Number Formula	Predefined pattern Formula dialog
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Frame	Define whether the rectangle should have a frame.	True False Formula	Yes No Formula dialog
Color	Frame color See Chapter 2.7.6. Colors		
Width	Width of the frame in the measuring system used in workspace	Number Formula	Formula dialog
Rounding	Rounding factor for the corners of the rectangle in %. 0% signifies angular 100% signifies: The corners of the rectangle are completely rounded.	Number Formula	Formula dialog

Property	Description	Value	Description
Shadow	Define whether the rectangle should have a shadow.	True False Formula	Yes No Formula dialog
Color	Shadow color See Chapter 2.7.6. Colors		
Pattern	Select one of a large number of predefined patterns. A number represents a pattern. This property will only be evaluated if "Filled" is set to "TRUE".	Number Formula	Predefined pattern Formula dialog
Width	Width of the shadow in the measuring system used in workspace	Number Formula	Formula dialog
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.6. Inserting Ellipses

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

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

5.6.1.Properties

Property	Description	Value	Description
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Filled	Define if the ellipse object should be empty or if it should be filled with a pattern and/or color.	True False Formula	Yes No Formula dialog
Color	Filling color See Chapter 2.7.6. Colors		
Pattern	Select one of a large number of predefined patterns. A number represents a pattern. This property will only be evaluated if "Filled" is set to "TRUE".	Number Formula	Predefined pattern Formula dialog
Locked	See Chapter 2.7.1. Locked		
Circle	Define if the ellipse should always be displayed as a circle centered in the.	True False Formula	Yes No Formula dialog
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Frame	Define if the ellipse should have a frame.	True False	Yes No

Property	Description	Value	Description
		Formula	Formula dialog
Color	Frame color See Chapter 2.7.6. Colors		
Width	Width of the frame in the measuring system used in workspace	Number Formula	Formula dialog
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.7. Inserting Picture Objects

Graphic files with the following formats can be integrated into your List & Label project:

File Extension	Description
.BMP	Bitmaps
.DIB	Device Independent Bitmaps
.EMF	Enhanced Meta Files
.JPG	JPEG-Standard
.PCD	Kodak Photo-CD
.PCX	Zsoft Paintbrush
.PNG	PNG-Standard
.RLE	Bitmaps
.SCR	Capture SCR
.TIF	Tagged Image File Format
.WMF	Aldus Meta Files

Note: Not every subtype of every format is supported.

To create a drawing or picture object, select **Objects > Insert > Picture** (STRG+D).

5.7.1.Properties

Property	Description	Value	Description
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Data source	Select how the data source for the graphic to be displayed is defined.	File name Formula Variable	Formula dialog
File name	Existing file name, will be evaluated if file name was selected in the property "Source". Select the desired graphic in the Windows familiar dialog. In this dialog, by using the option "Imbed in Project", you have the capability to imbed the graphic file into your project. The file is then copied	File name 🖼	File > Open dialog

Property	Description	Value	Description
	into your project and is available even though the file may be absent. In this case, embedded will be displayed instead of the file name.		
Formula	The file name is the result of a formula, if formula was selected in the property "Source". The result of this formula must be a value of the type "Drawing". You can also enter a valid file name, but it must first be converted to a value of the type "Drawing" with the function Drawing().	Formula √α	Formula dialog
Variable	The file name is the result of a variable, if variable was selected in the property "Source". Select the desired variable from the listbox. In this box you will find all variables of the type "Drawing" that were defined in your application.	List of all variables of the type Drawing	
Properties	Dependent upon your application, a further dialog can be opened in which more properties can be defined.	🖾	Opens dialog
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Keep proportions	Using this option, you define whether the graphic should be inserted so that the relationship between height and width is kept (True) or if the graphic should be adapted to fit the frame of the object (False).	True False Formula	Yes No Formula dialog
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.8. Barcode Objects

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

You can find a description of the barcode formats in Chapter **7.6. Barcodes**. To create a barcode object select **Objects > Insert > Barcode (**STRG+B).

5.8.1.Properties

Property	Description	Value	Description
(Contents)	Opens the "Contents" dialog	🖾	Opens dialog
Bar color	Color for the selected Barcode		

Property	Description	Value	Description
	See Chapter 2.7.6. Colors		
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Rotation	Rotates the object counter clockwise	0 1 2 3 Formula	0° 90° 180° 270° Formula dialog
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Optimum Size	Sets the optimum size for the Barcode. This property could affect size changes with the following formats: EAN 13 (all), DP-Leitcode, DP-Identcode, German Parcel, Postnet and FIM.	True False Formula	Yes No Formula dialog
Position	See Chapter 2.7.5.Position		
Font	Font for the Barcode text. Will only be evaluated if the "Show Text" property is activated. If the default is set the default font will be used. The font definitions dialog can be opened by using the button. You can set all properties in this property group in a single dialog.		Font dialog
Default	Default font will be used instead of the set values.	True False Formula	Default font No Formula dialog
Name	The name of the font can be selected. All fonts installed on the computer will be shown.	Font list	
Character set	Defines country interpretation of the character set. All character sets installed on the computer will be listed.	Number	Character set
Size	Font size in points. A list of available sizes for the selected font will be shown.	Number Formula	Predefined size Formula-Dialog
Width	Font width. 0 signifies standard width, otherwise the average character width will be displayed.	Number Formula	Width Formula dialog
Bold	Selects/deselects the text property "**Bold**".	True False Formula	Yes No Formula dialog
Italic	Selects/deselects the text property "**Italic**".	True False Formula	Yes No Formula

Property	Description	Value	Description
			dialog
Underlined	Selects/deselects the text property "**Underlined**".	True False Formula	Yes No Formula dialog
Strike Out	Selects/deselects the text property "**Strike Out**".	True False Formula	Yes No Formula dialog
Color	Font color See Chapter 2.7.6. Colors		
Show Text	Define whether the contents of the barcode should also be displayed as clear text.	True False Formula	Yes No Formula dialog
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.8.2. Barcode Contents
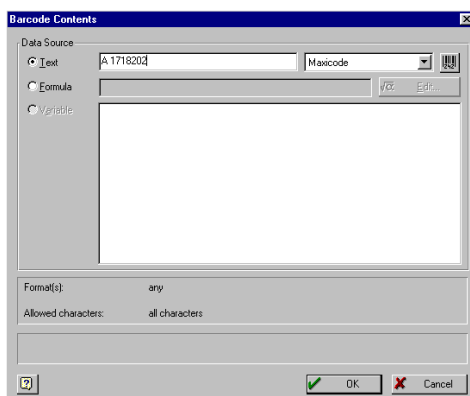
You can further define the barcode object using the contents dialog.

- If you wish to print fixed text in barcode form, select the option "**Text**". Enter the value to be printed in the first field. Select your desired type of barcode in the second field. Please note that certain requirements (number of characters, or spaces) must be met for the selected type of barcode. If the requirements are not met you will receive an error message. For some barcode types, for example Maxicode, additional options exist that can be edited in a further dialog. This dialog is opened with the button .

- If you wish to use a formula as a barcode, select the option "**Formula**" and define a valid formula expression using the "**Edit**" button. The return value of the formula must be of the "**Barcode**" type. You can also enter variables, but they must first be converted to variables of the barcode type using the function Barcode().

- If you wish to print a variable in barcode form, select the option "**Variable**". All variables of the type Barcode will be displayed in a listbox.

5.9. Insert Table objects

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

5.9.1.Properties

Property	Description		Value	Description
(Contents)	Opens the "Contents" dialog		🖹	Opens dialog
Appearance condition	See Chapter 2.7.3. Appearance Conditions			
Data lines				
Suppress	If the option "Data Lines Suppress" is selected, all data lines will be completely suppressed. This option is especially useful in combination with the option "Force Sums". The last option defines that the sum will be calculated when the data line is not printed. With the combination of both options and the use of groups and sum variables, the printing of interesting statistics is possible.		True False Formula	Yes No Formula dialog
Force Sums	Sums will be calculated even with the suppression of the data lines.		True False Formula	Yes No Formula dialog
Zebra Pattern	With the option "Zebra pattern" in the "Data Lines" group, you can assign alternating colors to the data lines. This will improve the clarity of large tables.		True False Formula	Yes No Formula dialog
	Color	Color of the selected pattern. See Chapter 2.7.6. Colors		

Property	Description		Value	Description
	Pattern	Pattern, with which every second data line is assigned. Select one of a large number of predefined patterns. A number represents a pattern.	Number Formula	Predefined pattern Formula dialog
Keep Together	If this property is activated, data lines will be kept together, so far as is possible, after a page wrap. The lines will be printed on the next page.		True False Formula	Yes No Formula dialog
Fixed Size	If the property "Fixed Size" is selected, the table will not automatically adjust it's size if less space is used than is available. If the property is not selected, the bottom edge of the table object will automatically move up to the last line. A footer line is separated by at least one line from the rest of the table when "Fixed Size" is selected. The empty line assumes the frame definition and layout of "Line Definition 1". Otherwise, the footer line will be located directly under the table. Please note: objects which may be linked to the table can only change their position automatically when "Fixed Size" is not selected.		True False Formula	Yes No Formula dialog
Separators Fixed	If this property is selected, the column separators will be continued from the last data line to the footer line. If this option is not selected, the column separators will reach only to the last data line. This option is only available for tables of "Fixed Size".		True False Formula	Yes No Formula dialog
Footer lines				
Keep Together	With the property Footer Lines "Keep Together", you can define,		True	Yes

Property	Description	Value	Description
Together	simultaneously with the option Lines "Keep Together", that footer lines will be kept together during a page wrap, as far as is possible.	False Formula	No Formula dialog
Locked	See Chapter 2.7.1. Locked		
Group Header Lines			
Keep Together	Using this option, you can define that group lines (intermediate headers) will not be separated from their data lines by a page wrap, if possible.	True False Formula	Yes No Formula dialog
Background Color	This property allows you to assign a background color to your table object. If the option is "TRUE", a background color can be selected with the option "Color".	True False Formula	Background color no background color Formula dialog
Color	See Chapter 2.7.6. Colors		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Frame Color	Color of the table frame See Chapter 2.7.6. Colors		
Font Default	With this property you can assign the default font for the table. The Font dialog can be opened using the appropriate button.		Font dialog
Page wrap Conditions	Additionally, you can enter a condition which will cause a page wrap to be performed as soon as the condition is met. If Page wrap = True, a page wrap will be performed after each line. If Page wrap = False, the page wrap will only be performed when necessary.	True False Formula	Yes No Formula dialog

Property	Description	Value	Description
Wrap before	See Chapter 2.7.4. Page wrap before Object Output.		

5.9.2. Table Contents

A table consists of various types of lines that can be defined independently of each other.

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

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

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

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

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

The individual lines types with their definitions and columns can be edited in the dialog "**Table Contents**".
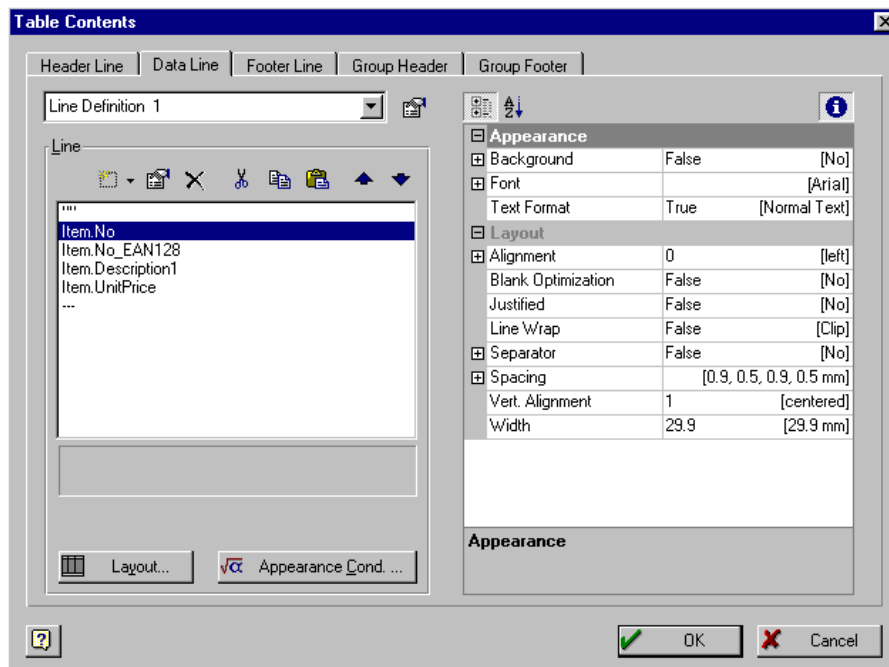
A card is provided for the definition of every line type and the columns.

5.9.3. Definition of Table Lines

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

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

If line definitions have been created, you can select the definitions to be used in the new line type.
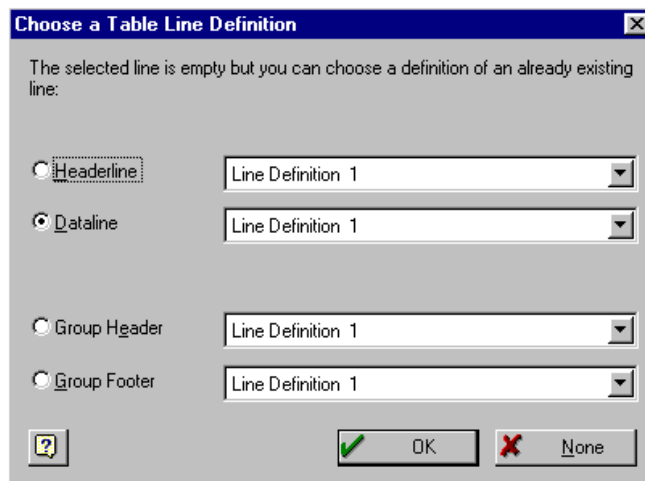
In the example pictured above, the definition for line definition 1 was selected for use in the data line.

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

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

5.9.4. Defining Line Layouts

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

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

To create multiple line layouts for a line, select the line definition that you would like to edit. If you have defined a layout for at least one line, you have the possibility of using the existing layout as a template for the new line layout.

As described in Chapter 5.9.6. Define Column Contents, you can now define the columns, their contents and properties for the new layout. Using the button "**Layout**" you can define the complete appearance for the new line layout. You can define, for example, special frames.

5.9.5. Editing the Line Definitions List

The Line Definitions list can be edited using the button 🖻.
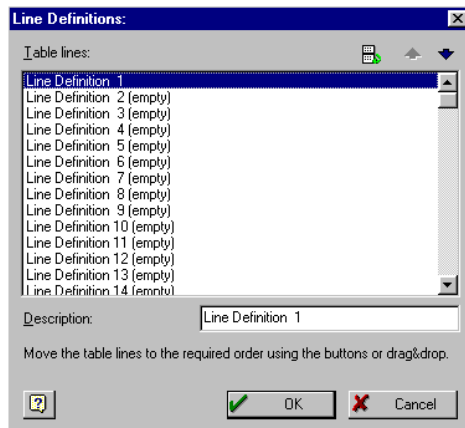
You can assign names to the line definitions. This will make it easier for you to find the line in complex layouts.

The order of the line definitions in the list can be changed using the arrow buttons or per drag & drop. Line definitions can be compromised, that means that empty line definitions will be moved to the bottom of the list, by using the 🖻 button.

5.9.6. Define Column Contents
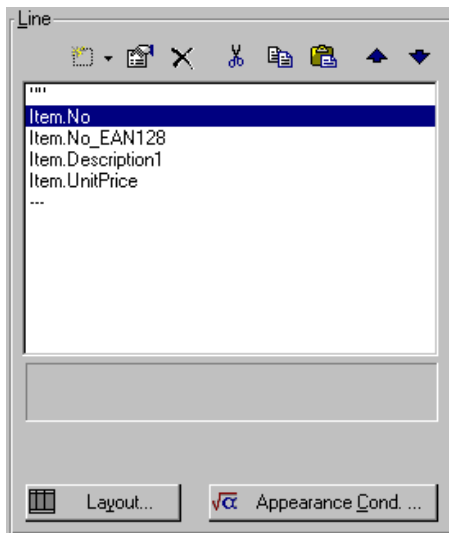
Table lines are column oriented, that means every line definition can contain many columns and every column can be separately edited and formatted.

Every column also has a definite type. The column types that are available are: Text, Drawing, Barcode, RTF-Text, Chart, HTML-Text and OLE Container.

Every column of a table object is represented by a line in the list. Multiple expressions or variables can appear within a column

With these buttons you can:

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

Editing of the column is done in the familiar Formula Editor, in which expressions can be defined as column contents, see Chapter 2.8. Variables, Formulas and Expressions for more information.

5.9.6.1. Format Lines

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

You can set the font preference for the complete table using the Select button. Newly inserted columns appear at the beginning in this font. Using the appropriate option in the Column Properties, you can select a different font for each column.
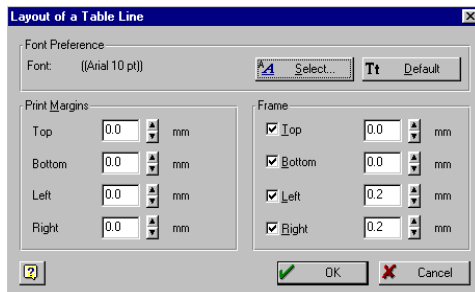
In the category "**Print Margins**", the margins that the table line should have within the table object can be set.

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

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

5.9.7. Appearance Conditions for Table Lines
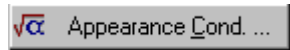
 Appearance conditions for the table lines can be assigned using the button "**Appearance-Cond**." Using this button opens the familiar dialog for the definition of logical expressions (see Chapter2.8. Variables, Formulas and Expressions).

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

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

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

5.9.8. Column Properties

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

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

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

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

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

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

Property	Description	Value	Description

Property	Description	Value	Description
Vertical Alignment	With this option you define the vertical alignment of the selected column.	0	Top
		1	Centered
		2	Bottom
		Formula	Formula dialog
Alignment (Text)	With this option, you define the text alignment. "Decimal" signifies that number values will be aligned at the decimal point.	0	Left
		1	Centered
		2	Right
		3	Decimal
		Formula	Formula dialog
Decimal-position	The position of the decimal point with the field, measured from the left corner of the preceding frame in mm.	Number	
		Formula	Formula dialog
Width	Width of the selected column in mm. If the sum of the column widths exceeds the width of the table, you will receive an error notice. The width of a field or column can also be changed in the workspace using the mouse by selecting the table object and moving the right border. The property list is recommended for precise resizing.	Number	
		Formula	Formula dialog
Background	The background color for the column can be defined.	True	Yes
		False	No
		Formula	Formula dialog
Color	Color of the selected pattern. See Chapter 2.7.6. Colors		
Pattern	Pattern to be assigned to the column. Select one of a large number of predefined patterns. A number represents a pattern.	Number	Predefined pattern
			Formula dialog

Property	Description	Value	Description
	represents a pattern. Default is the pattern 0, "transparent", no pattern selected.	Formula	dialog
Height (not for Text, RTF Text)	Height of the selected column in mm. The actual height of a table line is the height of the tallest column.	Number Formula	Formula dialog
Options (for Chart, HTML, OLE)	Opens the contents dialog for the appropriate object type.	📑	Opens dialog
Spacing	The distance between the frame and the contents of individual cells in a table can be set with the property "Spacing". The height of a table line with text objects is defined by the font size used and the top and bottom margins.		
Left	The distance between the left edge of the contents and the previous frame in mm	Number Formula	Formula dialog
Top	The distance between the upper edge of the contents and the previous frame in mm.	Number Formula	Formula dialog
Right	The distance between the right edge of the contents and the next frame in mm.	Number Formula	Formula dialog
Bottom	The distance between the lower edge of the contents and the next frame in mm.	Number Formula	Formula dialog

Property	Description	Value	Description
Separator	Insert a column separator on the left edge of the selected column. This property is not available for the first column.	True False Formula	Yes No Formula dialog
Thickness	Defines the thickness of the column separator. A Thickness of 0,0 signifies a thin hair line. If you do not wish to have separators, the option can be deselected with the option "Separator".	Number Formula	Formula dialog

5.9.9. Defining Footer Lines

It is often desirable to print footers along with your table. These are considered to be normal table lines with the exception that footers always appear at the end of the table on each page. If you want the footers to be printed on the last page only, assign the appearance condition LastPage(). Additional information about this function can be found in Chapter 7.4. List of Available Functions.

5.9.10. Defining Group Lines

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

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

The assigned column definition for the group line could be: "Letter: «Left$(NAME,1)»" or "Letter: "+Left$(NAME), depending on which modus you are in. The syntax assistant will automatically display the proper syntax. For every new beginning letter in the NAME, the text "Letter: " followed by the letter will be printed.

More than one line layout is also possible. Hierarchically indented intermediate headers can be created. You can define a line layout, like in the above example, which will generate an intermediate header using the first letter of the variable NAME. Additionally, you can define a second line layout which generates an intermediate header using the first two

characters of the variable NAME. The expression for this would be "Left$(NAME,2)", a suitable column definition would be "Letters: «Left$(NAME,2)»". For this second layout, special margins can be set using the button "Layout", for example "left" = 10.0 mm. The result could look like:

.....

Letter: A

Letters: AA

...

data line

...

Letters: AB

...

data line

...

...

Letter: B

Letters: BA

...

data line

...

Letters: BB

...

dataline

...
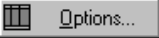
Appearance Condition
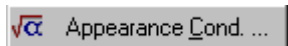
√α Appearance Cond. ...   It is also possible to use appearance conditions for group header lines. Further information can be found in the section "Appearance Conditions for Table Lines" in this chapter.

Options for Group Header Lines

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

Select the Sum Variable(s) that should be returned by the group header line. This setting is useful to create group sums, for example to add the price of all items of a certain item group. See also Chapter 4.3. Sum Variables.
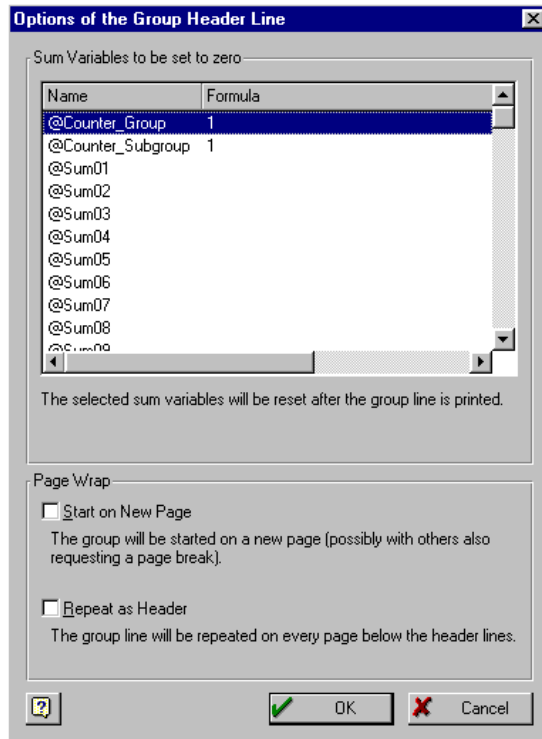
Select the option "**Start on New Page**" if the header line should cause a page wrap, i.e. if each group should start on a new page. If multiple group lines are to be printed with this option at the same time, they will be printed under each other on the same page.

The option "**Repeat as Header**" prints the group header on the next page after a page wrap.

5.9.11. Define Group Footers

A further special line type is the group footer. In principle, it works in exactly the same way as the group header, but appear only after the conditions have changed.
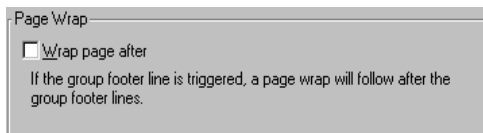
As "**Wrap Option**" only "**Wrap Page After**" is available for group footers, this means that a page wrap will be performed after the printing of the group footers.

5.10. Formatted Text

With the formatted text object you can, in contrast to normal text objects, change the format within a line. In addition you can also use variables in these objects.

When should you use the formatted text object and when the text object? Normal text objects should be given preference for every day jobs, since they contain less information and they are, therefore, printed faster. Consequently you should use a formatted text object in case you either can not realize a special format type with the normal text object or succeed only with great effort.

5.10.1. Properties

Property	Description	Value	Description
(Contents)	Opens the Contents		Opens dialog
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Rotation	Rotates the object counter clockwise. This function is only available on Windows NT based systems (Windows NT, Windows 2000, Windows XP).	0 1 2 3 Formula	0° 90° 180° 270° Formula dialog
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Page wrap	Defines if the object can cause a page wrap. If this property is activated, the contents of the object that exceed it's size will be automatically wrapped to the next page. This is interesting for RTF-Objects that should use multiple pages. With labels, the next label will be started only when all objects with this option have been printed on the previous label. This option may not be available if page wrapping is not supported by your application. The option is activated by default. If you wish to print an object on every page of a list project, this option must be deactivated or the object will only be printed once per project.	True False Formula	Yes No Formula dialog
Wrap before	See Chapter2.7.4. Page wrap before Object Output.		

5.10.2. Text Contents

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

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

Select the font, size, color and character set.

B *I* U S x² x₂	bold, italic, underlined, strike out, superscript, subscript
≡ ≡ ≡ ≡ 🎨	Left justified, centered, right justified, block and text background color. System requirement for block and background color: Windows 95 with Rich Edit 3.0, Windows NT 4.0 with Rich Edit 3.0, Windows Millennium Edition or Windows 2000, XP
⊫ ⊨ ⊩ ≔	Indent paragraph (2.5 mm), right Mouse click: indent following lines Unindent paragraph (2.5 mm), right Mouse click: unindent following lines Set line and paragraph spacing Enumeration
📋 ↺ ↻	Insert from clipboard Undo action Repeat action
√α 👤	Open the Formula Wizard Insert OLE-Object. You can also insert using the clipboard.
☑ WYSIWYG-Wrap.	Word wrapping according to object size. This is only an approximate display and minimal differences may occur.

5.10.3. The Context Menu

A context menu is available in the editor, by which you can open various formatting possibilities for characters and paragraphs.

The RTF-Object also supports so called "**Embedded Objects**", objects that are embedded into the text, for example graphics. These objects can be inserted using the clipboard from, for example, MSPaint.

5.10.4. ToRTF

An important function of the Formula Wizard is ToRTF, which you can use to convert variables into RTF-text.

This could have the following appearance:

«ToRTF$(Recipient.Salutation)» «ToRTF$(Recipient.Lastname)»

5.11. Inserting Chart Objects

By selecting **Objects > Insert > Chart** or clicking the respective icon you can insert a chart object. This object is used for analyzing

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

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

5.11.1. Properties

Property	Description	Value	Description
(Content)	Opens the contents dialog	📑	Opens dialog
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Wrap before	See Chapter2.7.4. Page wrap before Object Output.		
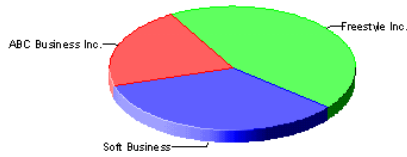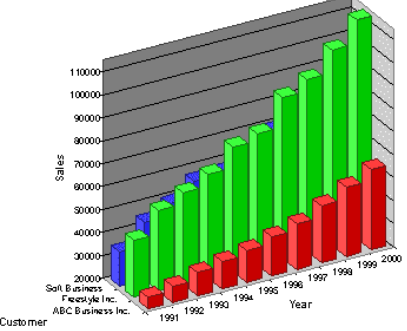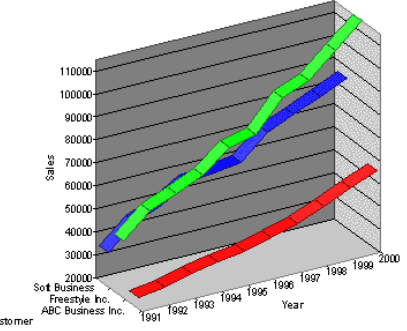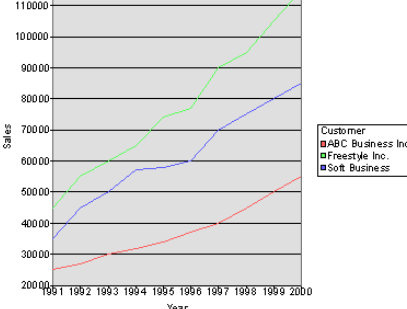
5.11.2. Selecting the chart type

Double click a chart object to reach to the property dialog.
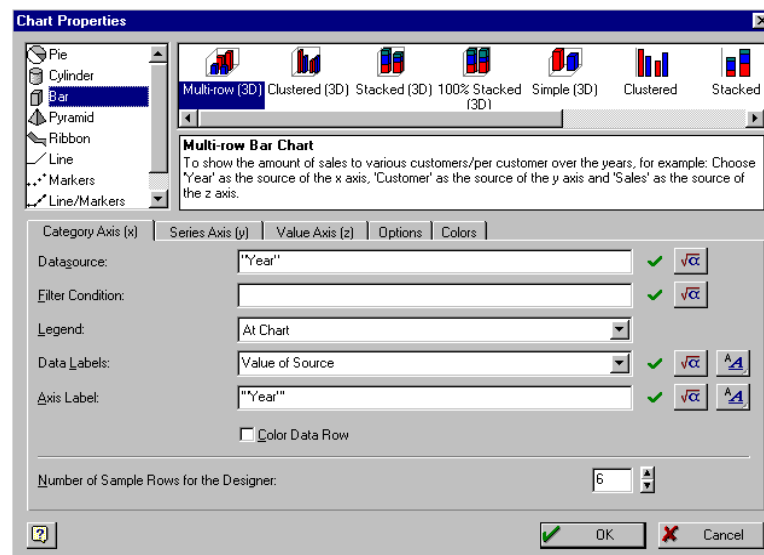
Select the chart type from the list on the left hand side. The following types are available:

	Pie Chart: a typical pie chart.
	Cylinder/Bar/Pyramid: various types of bar charts, that differ mainly in the form of the bars.
	Ribbon: a three dimensional ribbon chart.
	Line/Symbols/Line & Symbols: two dimensional charts.

5.11.3. Selecting the sub type

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

Sub types for cylinder, bar and pyramid charts:

	Multi row: A "normal" 3D chart, showing the sales per month and per employee, for example
	Simple 3D/Simple: Only two axes are available, showing the total amount of sales per month for example. The simple type is without the 3D effect.
	Clustered 3D/Clustered: Values on the x axis are grouped allowing a direct comparison of values. The clustered type is without the 3D effect.
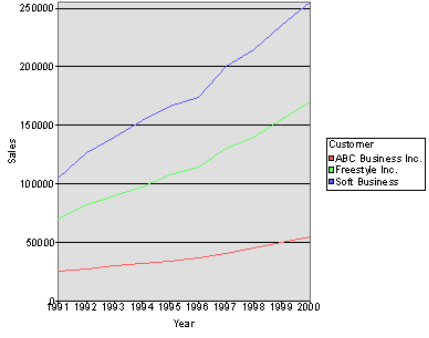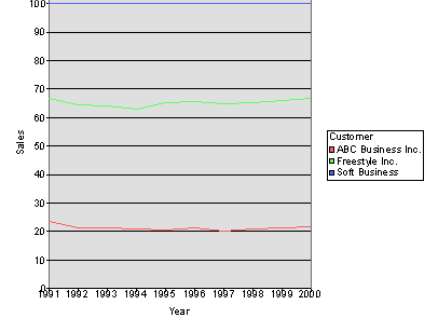
Stacked 3D/Stacked: This chart type is designed to show the share of each contribution, e.g. the share of sales each employee contributes to the total. This type is not available for pyramid charts. The stacked type is without the 3D effect.

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

Sub types for line and symbol charts

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

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

5.11.4. Chart properties

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

Pie charts

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

Options for the data source

Data source	Select the data source for the segments, e.g. "Name" for people, "Month" for date values, and so on.
Filter condition	If you like you can define a filter condition. Only the records matching the condition will be used for the chart data (e.g. `Left$(Name,1) = 'A')`.
Legend	Select the position of the legend. If you select "at chart" the values are given directly at the segments.
Segment labels	Select the text which should be used to label the legend. Some preconfigured values are available, e.g. "Value and Percentage without decimals". Alternatively you may enter your own formula, select "Formula..." from the list in order to do so. By clicking on ![icon] you may also enter a formula. The ![icon] allows you to set the font for the labels. A left click opens a font dialog, a right click resets the font to the default object font.
Color data row	The segments are colored with different colors in order to make the diagram easier to read.
Number of sample rows for the Designer	The Designer has no access to the "real" data which will appear in your chart when printing. In order to have a picture of how your chart is going to look like, you can set the number of segments to appear in the Designer here.

Options for the type of calculation

Calculation mode for equal values	Specifies the handling of equal source values. An example: you want to have the sum of sales to your customers. In this case, select "sum of values". If you're interested in the average price of sales to a customer, select "mean value". If only the number of sales to a customer is of interest, select "number of values". As the actual value is not of interest in this case, the source field will be deactivated.
Source for segment size	Select the data source for the segment size, e.g. "Value", "Price" etc.
Minimum share	Especially when you have numerous values that make up small contributions it may be desirable to summarize these in a common segment. Select the maximum value up to which the segments will be summarized.
Summarize lower values as	Defines the label for the summarized segment. If you do not enter your own formula, the label will be adapted to the "normal" segment label layout on the data source card.

Other chart types

Options for category and series axis

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

Data source	Select the data source for the row, e.g. "Name" for people, "Month" for date values, and so on.
Filter condition	If you like you can define a filter condition. Only the records matching the condition will be used for the chart data (e.g. `Left$(Name,1) = 'A')`.
Legend	Select the position of the legend. If you Select "at chart" the values are given directly at the axis. Otherwise, a legend will be added on the left, right, bottom or top of the chart.
Data labels	Select the text which should be used to label the legend. Some preconfigured values are available, e.g. "Value of source". This setting will display the source value on the axis, i.e. the name of a customer, the month,... Alternatively you may enter your own formula. Select "Formula..." from the list in order to do so. By clicking on $\sqrt{\alpha}$ you may also enter a formula. The $^A\!A$ allows you to set the font for the labels. A left click opens a file selection dialog, a right click resets the font to the default object font.
Axis label	Select a text for your axis label here. By clicking on $\sqrt{\alpha}$ you may also enter a formula. The $^A\!A$ allows you to set the font for the labels. A left click opens a file selection dialog, a right click resets the font to the default object font.
Color data row	This attribute may only be selected for either the series or the category axis. The selected data row will then be colored in altering colors.
Number of sample rows for the designer	The designer has no access to the "real" data which will appear in your chart when printing. In order to have a picture of how your chart is going to look like, you can set the number of values to appear in the designer here.

Options for the value axis

Calculation mode for equal coordinates	Specifies the handling of equal source coordinates. An example: you want to sum the sales to customer Smith in January. In this case, select "sum of values". If you're interested in the average price of your sales to this customer, select "mean value". If only the number of sales to the customer is of interest, select "number of values". As the actual value is not of interest in this case, the source field will be deactivated.
Data source	Select the data source for the value axis, e.g. "Value", "Price" etc.
Range	You can limit the shown data range in order to e.g. regard huge divergences within the values. If your values have high peaks, you can cut them off by setting a max limit. If you leave the default setting "automatic" the chart will display all values.
Legend, Coordinate labels, Axis label	These options are the same as for the series and category axis. The legend can only be displayed at the chart or not at all, however.

Options for the chart object

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

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

Color options

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

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

	chart, this is the color of the bars, for a pie chart the segment color. To select a color, select it from the upper Combobox. The "..." entry brings up a standard color selection dialog.

5.11.5. Examples

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

Multi row bar chart

This would be the simplest way to analyze your data; you'd have a diagram showing the total transaction volume for each month and customer:
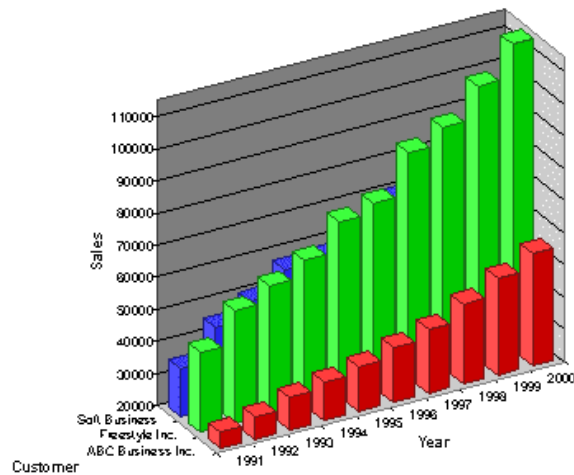
...this is how it's done:

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

Pie chart

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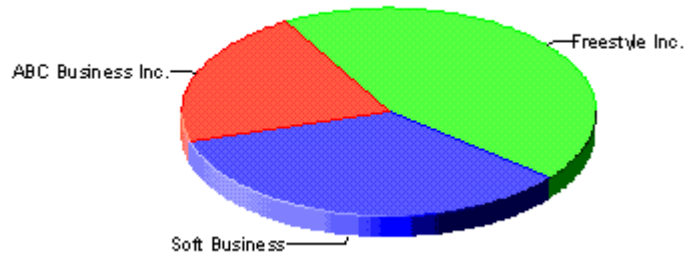
...this is how it's done:

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

100% stacked bar chart

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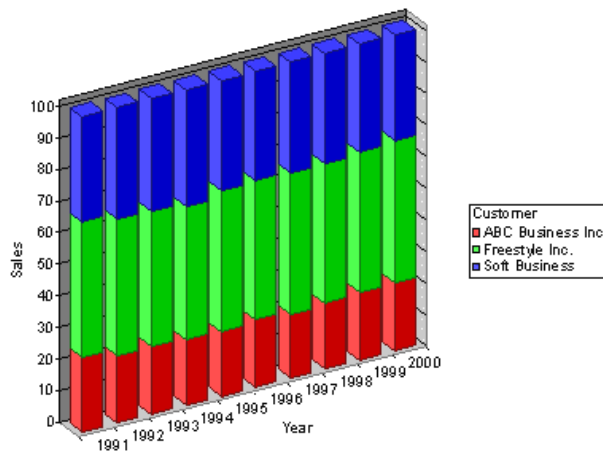
...this is how it's done:

Insert a new chart object, select "Bar" as type and "100% stacked" as sub type. Select "Month" as the data source for the category axis and "Customer" as the data source for the series axis. For the value axis,

select "Sales". Edit the axis labels and title as required – the procedure is exactly the same as for the multi-row bar chart. This is why you can simply switch between the two sub types without having to reconfigure your data sources.

5.12. HTML formatted text

To display HTML formatted text (e.g. web sites) within your projects, you can use the HTML text object. Please note that due to license restrictions no GIF files can be displayed.

The object supports the HTML 3.2 specification. Partially supported are some extended tags as well as cascading style sheets.

To insert a HTML text object use the toolbar or **Objects > Insert > HTML-Text**.

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

5.12.1.Properties

Property	Description	Value	Description
(Contents)	Opens the "Contents" dialog		Opens dialog
Appearance condition	See Chapter 2.7.3. Appearance Conditions		
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Wrap before	See Chapter 2.7.4. Page wrap before Object Output		

5.12.2. HTML-Contents

You can define the contents of the object in the HTML-Object-Properties dialog.
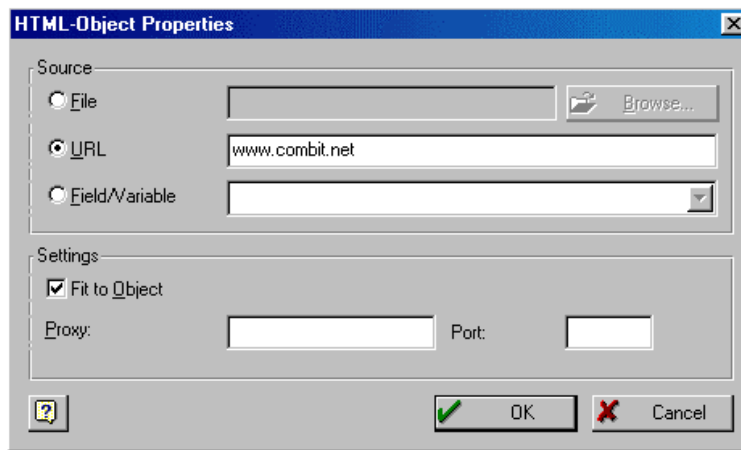
- File – select this option to display the contents of a previously saved HTML file. The "**Browse**..." button opens a file selection dialog. The file must be located on a local storage medium or network.
- URL – this option is used to display websites (e.g. www.combit.net). The contents are downloaded at runtime, thus you'll need a connection to the Internet.

If your access is via a proxy server, which is the common case in companies, you must enter its address in the settings group. By default, your current proxy configuration will be set. Leave this setting in order to be able to distribute your projects to other systems with a different proxy configuration. The current configuration will always be taken in this mode.

- Field/Variable – if your application makes HTML contents available, these can be selected here. Refer to your application's documentation for details.

- Fit to Object: Select this option to fit the contents into the object. If the option is deactivated, the contents are fit to the object width and output may wrap over several pages.

5.13. Insert OLE-Server Documents

The OLE Container is available for the insertion of OLE-Server documents into your project. This gives you the capability to embed and print documents that were created with other applications, for example Word, Excel, Visio or MapPoint, into your project without changes.

You can create an OLE-Object using **Objects > Insert > OLE Container**. After defining the size and position of the object on the

workspace, the standard "Insert Object" dialog will open. You can select the object type and select either "Create New" or "Create from File".

You can select, for example, an existing Excel file and insert it into the object frame.

Note: Only the first Page of an embedded document can be displayed.

5.13.1.Properties

Property	Description	Value	Description
(Contents)	Opens "Contents" dialog		Opens dialog or host application
Appearance Condition	See Chapter 2.7.3. Appearance Conditions		
Locked	See Chapter 2.7.1. Locked		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Wrap before	See Chapter 2.7.4. Page wrap before Object Output		

5.13.2. Contents

If contents are available the host application will be opened, otherwise the standard "Insert Object" dialog.

5.14. Inserting Form Templates

Form templates are scanned bitmaps of actual forms that you can place in the background of your workspace to aid in the exact placement of objects. The templates are displayed on the workspace but will not be printed and cannot be edited.

To insert a form template in the background of your workspace, use **Objects > Insert > Form Templates**.

5.14.1.Properties

Property	Description	Value	Description
File name	Select the file that contains the required form template. The same formats are supported as in graphic files. (See Chapter 5.7. Inserting Picture Objects).	File name	Open file dialog
Fade color	The selected color will be added to the template to fade the color. See Chapter 2.7.6. Colors		
Name	See Chapter 2.7.2. Name		
Position	See Chapter 2.7.5.Position		
Keep proportions	Using the option "Keep Proportions" you can define whether the graphic should be	True False	Yes No

Property	Description	Value	Description
	inserted in the correct relationship between height and width (True), or if the graphic should be resized to occupy the complete object.(False).	Formula	Formula dialog
Pagewrap before	See Chapter 2.7.4. Page wrap before Object Output		

Tip: When using a template to create your own form or to fill out a form, size and position of the template are of critical importance. We recommend that the template be placed using the Property List, as this allows a more exact placement than the use of the mouse.

After creation, select the form template using the tool window Objects. The template can no longer be selected in the workspace.

5.15. Arrange

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

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

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

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

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

5.15.1. To Front

All selected objects are moved on top of the stack of sheets, that is, they have priority over the objects lying deeper. Use the command **Objects > Arrange > To Front** to bring the selected object to the top of the stack, or use the shortcut CTRL+POS1.

5.15.2. To Back

Puts the selected objects to the bottom of the stack of sheets. Use the command **Objects > Arrange > To Back** place the selected object at the bottom of the stack, or use the shortcut CTRL+END.

5.15.3. One Forward

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

5.15.4. One Backward

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

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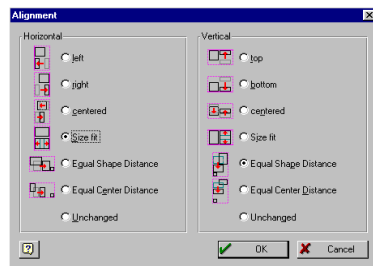
Using the command **Objects > Arrange > Alignment** you can align multiple objects in relation to each other. A minimum of two objects must be selected to use this command.

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

Left / Right / Top / Bottom

 The functions reposition the selected objects at the corresponding border of the selection rectangle. They will not change in size.

Centered

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

Size Fit

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

Equal Shape Distance

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

Equal Center Distance

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

Unchanged

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

5.16. Group

You can combine objects to a group. This will cause them to be selected as one object. This again means all operations (resizing, moving,...) will take place on all member objects of this group. An object can only be a member of one group, thus groups cannot be placed in other groups.

To group objects, mark them and select the command **Objects > Group** (also accessible from the context menu), or the shortcut CTRL+G.

To undo a grouping, select the group and select **Objects > Ungroup**.

5.17. Assign to Layer

You assign the currently selected object(s) to one of the available layers. See Chapter 2.6. Working with Layers.

5.18. Copy to Layer

You can not only assign objects to layers, but objects can be copied to another layer using the command **Objects > Copy to Layer**. The source object will retain it's layer assignment and a copy will be placed into the target layer.

5.19. Contents

With **Content** (or ENTER) the Contents dialog of the selected object will be opened, if available. See Chapter 5. Menu Objects for more details.

5.20. Properties

The command **Properties** in the menu **Objects** brings the "Properties" tool window to the foreground if it is docked behind another tool window.

The properties of the individual objects have been described in detail in the appropriate subchapter of Chapter 5. Menu Objects.

5.21. Locked

You can lock/unlock the object using the object property "Locked". See Chapter 2.7.1. Locked for a detailed description.

5.22. Object List

By using the command **Objects > Object List** or the shortcut NUM* (key * in the number pad) a dialog with a list of all available objects is opened. This dialog contains the Object List, the appropriate to window, and features added functionality.
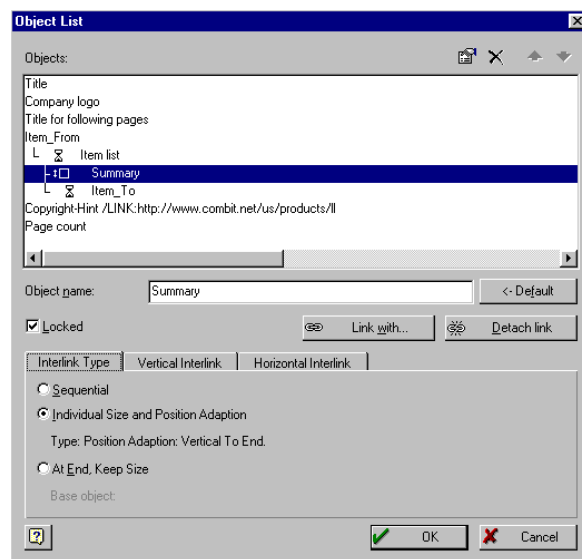
Editing or Deleting Objects

This can be done using the "**Edit**" and "**Delete**" buttons. You may also start the object properties dialog by double-clicking on an object in the list. If multiple objects are selected when clicking the "**Delete**" button, all objects are deleted.

Lock objects

If the option "Locked" has been selected for an object, the object can be neither selected nor edited on the workspace. It can, however, be edited using the Object List. If the option is not selected, the object can be selected and edited on the workspace.

Naming Objects

The edit control allows you to change the name of the selected object. The name is of course the same name that can be changed using the property list.

Resetting the Name

The "<-Default" button will set the default text (type and coordinates) as the name of the object.

5.22.1. Linking Objects

By linking objects you can make sure that one object is printed after another (either in time or space). A linked object may also adapt its size to the parent object (ex. table). Keep in mind that tables can not be linked to each other.

The linked objects are placed in a "parent-child" hierarchy. The object, that other objects are linked to, becomes a parent object. The objects linked to a parent object are child objects.

Click this button to link objects to the selected object in the object list. A list of available objects in your project is displayed. Select the object that you want to link to the parent object. The links are shown in the Object List in Windows Explorer style. You can easily tell which object is the child and which object is the parent object. To link more than one object to the parent object, repeat the steps described above.

5.22.1.1.   Interlink Type
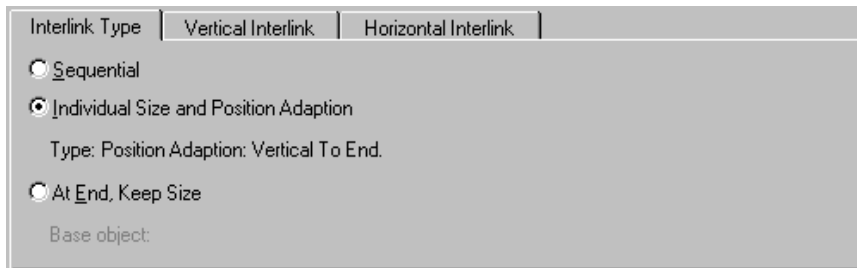
Once you select a linked object, the type of link can be set in the **Interlink Type** card. The options are:

Sequential

A sequential link means the child object will be printed after the parent object. This is useful, for example, if you want to state the number of records on the current page in a table. This number, of course, is only known after printing the table. By linking the counter field sequential to the table, you ensure the correct result to be printed.

The sequential link is the default setting after linking two objects. It is symbolized by an hourglass icon in the object list.

If a sequential link is selected, none of the options for adjusting position or size in the interlink type group are set.

Link in Space

A link can also affect the position and size of the linked object. **A size and position adaptation implies automatically a sequential link.** If the parent object is smaller than the maximum space available, the linked object can use this to move relative to that change to have a constant distance to the lower edge of the parent object.

For example, if you link an object to a table object using "**Position adaptation: Vertical, to end**", the object will have a constant distance

to the footer line of the table and will "float" along with it when the size of the table changes.

These settings will only affect printing as they depend on the change of the object's size due to the data that is being printed. Changes in the workspace on the parent objects do not have an effect on the child objects.

The symbol for this type of link is a rectangular shape with one or more arrows.

The selected link type is displayed on the Interlink Type card.

Individual Size and Position Adaptation

The link type will be selected over the cards Vertical Interlink and Horizontal Interlink. In each case 2 different types of link mechanisms are available:

Link Type: Position Adaptation: means that the position of the child object will adapt to the changes of the parent object's position.

- Option "**Relative to Begin**": the child object moves like the top left corner of the parent object.
- Option "**Relative to End**": the child object moves like the bottom right corner of the parent object.
- Option "**To End**": the top line of the child object begins at the end of the parent object independent of the original position. This causes an implicit adaptation in size on the first page the child object is printed on.

Link Type: Size Adaptation: means that the size of the child object will adapt to the changes of the parent object's size.

- Option "**Proportional**": the size of the child object changes in relation to the size of the parent object. If the parent object will be 1 inch smaller the child object will be 1 inch smaller.
- Option "**Inverse**": the size of the child object changes in the opposite direction: if the parent object shrinks by 1 inch, the child object increases in size by 1 inch.

Horizontal Interlink

The child object adapts the position respective to the width of the corresponding changes of the parent object. Which corner is the decisive one depends on the selected option.

Vertical Interlink

The child object adapts the position respective to the height of the corresponding changes of the parent object. Which corner is the decisive one depends on the selected option.

If both options "**Horizontal**" and "**Vertical**" are set, the child object adapts to both changes of the parent object.

Keep Size

If you want to link an object to another object so that the child object keeps its size but adapts its position to start under the parent object select the link type "**At End, Keep Size**". In contrast to the plain position adaptation this option is related to available space. If necessary a page wrap is done and the child object starts on a new page. You can use this option to place multiple objects (pictures, charts, etc.) underneath each other. In this case the "**base object**" is the first object in the hierarchy with activated page wrap.

Detach link
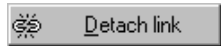
 Using this button you can unlink the selected object. It then is available again as single object in the object list.

5.22.1.2. Examples

As stated above, links in time should be used whenever the contents or conditions set in the child object are not available or met before the parent object is printed. It is, for example, inevitable to use a link in time to use the function LastPage(). If you want to print the total price on the last page of an invoice, you may insert a text object with the appearance condition LastPage() and link it in time with the table containing your items. If the link was omitted, the object would never be printed, as variables (text objects) are printed before fields (tables) by default, and the total price is not available at that time, nor whether the current page is actually the last page.

A common use for links in space is the printing of invoices. An invoice usually consists of a table, which holds the items of the invoice. The first page, of course, should be different from the others, because it contains the invoice header with your address, logo etc. By using an inverse, vertical size adaptation, you get by very easy: Design an invisible frame on the base layer (see chapter 5.4. "Inserting Rectangles"), which begins where the invoice table should start, and ends where the invoice table should start on page one. Select "Page()=1" as appearance condition for the rectangle and link the table (child) to the rectangle (parent). Select an inverse, vertical size adaptation and a position adaptation vertical, relative to end. In this way you solve your problem with just one table and don't have to update both tables if you change your table design later on.

A typical example for a "**keep size**" link is a chart object that should be printed at the end of a table object. In most cases you know what size the chart should have. Place the chart object in the desired size an link it to the table object. Check the options   "**At End, Keep Size**". Independent of the final ending of the table object the chart object is printed under the table object in the desired size, if necessary on a new page.

6. View

The View Menu offers different options for the appearance of the Designer.

6.1. Zoom Levels

To work precisely in complex projects, various zoom modes are available:

Full (Full view, no Zoom)	ALT+1
Zoom 2 Times	ALT+2
Zoom 4 Times	ALT+3
Zoom 8 Times	ALT+4

The current zoom mode is displayed with a ✔ in the menu.

6.2. View Mode

With the options "**Layout**", "**Layout Preview**" and "**Preview**" you can select the view mode.

6.3. Windows

With this command you can turn tool windows, rulers and toolbars, on and off. In this way, you can remove the Preview Window, the Variable List, the Object Window, the Properties List, and the Layers Window to create more room on your workspace. Active windows are displayed in the menu with a ✔ .

7. Appendix

7.1. Working with Functions

7.1.1. Function Syntax

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

Result = **Function(Arguments)**

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

List & Label evaluates the expression and replaces the Function(Arguments) with the "return value".

This means, when you enter a

Function(Argument)

the

Return value

is generated.

Definitions:

Element	Description
Function()	The name of the function in it's correct syntax. The parentheses for the arguments are a part of the function name. Even if the function contains no arguments, the parentheses are required.
Arguments	Values used in the expression to calculate the return value. The arguments are entered in the parentheses without spaces. A function can have no, one, or multiple arguments. Often, functions require arguments of a specific value type (see below). It is important that the argument has the required value type.
Return value	The value List & Label calculates from the function and the arguments. The value type that is returned is dependent upon the value type of the function and argument.

7.1.2. Value Types

Type	Meaning
BOOLEAN	Logical values TRUE or FALSE. If the condition is met, the value is TRUE, otherwise FALSE.
String	Any combination of letters and numbers. If this is used as fixed text, it must be in single or double quotation marks so that they can be distinguished from variables.
DatE	A number representing the date using a Julian calendar.

NUMBER	A string that consists of the digits 0-9, the decimal point and the minus (-) sign. Other characters are not allowed. Strings of this type need not be enclosed in quotation marks ("").
Barcode	A string used as barcode text.
PICTURE	A supported graphic format.
RTF	A formatted text

7.1.3. Examples for the Use of Functions

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

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

The function "FStr$(NUMBER, STRING)"

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

The format string can contain the following characters:

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

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

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

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

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

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

Value	Result
0	" 0"
1	" 1"
255	" 255"

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

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

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

You want the total price? No problem:

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

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

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

Some additional examples:

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

The function "LastPage()"

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

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

116

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

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

The function "Left$(STRING, NUMBER)"

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

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

7.2. Order of priority

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

The order of priorities is

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

7.2.1. Operators

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

Operator	Meaning	Data Types	Result Types
>	Greater than	STRING, NUMBER, DATE, RTF	BOOLEAN
> =	Greater than or equal	STRING, NUMBER, DATE, RTF	BOOLEAN
<	Less than	STRING, NUMBER, DATE, RTF	BOOLEAN
< =	Less than or equal	STRING, NUMBER, DATE, RTF	BOOLEAN
=	Equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN

= =	Equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
<>	Not equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
! =	Not equal	STRING, NUMBER, DATE, RTF, BOOLEAN	BOOLEAN
%	Modulo	NUMBER	NUMBER
+	Add	STRING, NUMBER, DATE	STRING, NUMBER, DATE
-	Subtract	NUMBER, DATE	NUMBER, DATE
*	Multiply	NUMBER	NUMBER
/	Divide	NUMBER	NUMBER
AND	Logical AND	BOOLEAN	BOOLEAN
OR	Logical OR	BOOLEAN	BOOLEAN
XOR	Logical exclusive-OR	BOOLEAN	BOOLEAN

7.3. List of the List & Label Variables

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

7.3.1. List of Variables

Name	Description
LL.SortStrategy	Selected sorting can be set, dependent upon the application in the **Project** Menu.
LL.FilterExpression	Selected project filter, can be set in **Project-** Menu.
LL.OutputDevice	Output device. Can be used to assign object special formats for certain output formats (HTML, RTF, PDF,…)
LL.CountData (not for List projects)	Number of the transmitted records. This number also contains the records that were not printed due to filter conditions. This number will increase with every record.
LL.CountDataThisPage (not for List projects)	Number of the transmitted records on the current page. This number also contains the records that were not printed due to filter conditions. This number will increase with every record.
LL.CountPrintedData (not for List projects)	Number of the actually printed records.
LL.CountPrintedDataThisPage (not for List projects)	Number of the actually printed records on the current page.

LL.Device.Page.Size.cx	Physical page width of the output device in project coordinates. Can be used in formulas to fit objects to larger output formats. Example: Set the property Position.Left in the Properties List to the value 0, the property Position.Right to LL.Device.Page.Size.cx. The object will occupy the total page width.
LL.Device.Page.Size.cy	Physical page height of the output device.
LL.Device.Page.Name	Name of the output format (example "A4").
LL.Device.PrintableArea.Size.cx	Printable page width of the output device.
LL.Device.PrintableArea.Size.cy	Printable page height of the output device.
LL.Device.PrintableArea.Offset.cx	Width of the non-printable border in project coordinates. Example: In the Properties List, set the property Position.Left to the value LL.Device.PrintableArea.Offset.cx. The object will always lie exactly on the left edge of the printable area of the printer.
LL.Device.PrintableArea.Offset.cy	Height of the non-printable border in project coordinates.
LL.Device.Name	Name of the output device(printer).

7.3.2. List of Fields

Name	Description
LL.FcountData	Number of transmitted records. This number also includes the records that were not printed due to filter conditions.
LL.FcountDataThisPage	Number of transmitted records on the current page. This number also includes the records that were not printed due to filter conditions.
LL.FcountPrintedData	Actual number of printed records.
LL.FcountPrintedDataThisPage	Actual number of printed records on the current page.

7.4. List of Available Functions

In the following table you will find all functions, listed alphabetically, available in List & Label. In the left column the function names, in the middle column the number and permitted value types of the arguments and in the right column, the value type of the return value.

Function	Arguments	Return values
Abs	Number	Number
AddDays	Date, Number	Date
AddHours	Date, Number	Date
AddMinutes	Date, Number	Date
AddMonths	Date, Number	Date
AddSeconds	Date, Number	Date

Function	Arguments	Return values
AddWeeks	Date, Number	Date
AddYears	Date, Number	Date
Asc	String	Number
AskString$	String, Boolean, String, Number	String
Atrim$	String	String
Barcode	String, String	Barcode
Barcode$	Barcode	String
BarcodeType$	Barcode	String
Case$	Number, String [,String]	String
CheckMod10	String	Number
Chr$	Number	String
ChrSubst$	String, String [,String]	String
Cond	Boolean, All, All	All
Contains	String, String	Boolean
Continued	-	Boolean
Cstr$	Number, String	String
Date	String	Date
Date$	Date, [String, [String]]	String
DateHMS	Number, Number, Number	Date
DateInRange	Date, Date, Date	Boolean
DateToJulian	Date	Number
DateYMD	Number, Number, Number	Date
Day	Date	Number
Day$	Date	String
Dow$	Date	String
Dow	Date	Number
Drawing	String	Drawing
Drawing$	Drawing	String
Empty	String	Boolean
Even	Number	Boolean
Exp	Number	Number
Exp10	Number	Number
Frac	Number	Number
FStr$	Number, String	String
Hour	[Boolean]	Number
Hyperlink$	String, String, Boolean	String
If	Boolean, All [,All]	All
Int	Number	Number
JulianToDate	Number	Date
LastPage	-	Boolean
Left$	String, Number	String
Len	String	Number
Locale$	Number[,String]	String
LocCurr$	Number[,String]	String
LocCurrL$	Number[,String]	String
LocDate$	Date, [,String, [Number]]	String
LocNumber$	Number[,String]	String
LocTime$	Date, [,String, [Number]]	String

Function	Arguments	Return values
Log	Number	Number
Log10	Number	Number
Lower$	String	String
Ltrim$	String	String
Max	Number | Date	Number | Date
Mid$	String, Number [,Number]	String
Min	Number | Date	Number | Date
Minute	[Boolean]	Number
Month	Date	Number
Month$	Date	String
Now()	-	Date
NumInRange	Number, Number, Number	Boolean
Odd	Number	Boolean
Ord	String	Number
Page	-	Number
Pow	Number, Number	Number
ProjectPath$	-	String
Rep$	String, Number	String
RGB	Number, Number, Number	Number
Right$	String, Number	String
Round	Number, Number	Number
Rtrim$	String	String
Second	[Boolean]	Number
Sqrt	Number	Number
Str$	Number | Date [,Number [,Number]]	String
StrPos	String, String [,Number]	Number
StrRPos	String, String [,Number]	Number
StrSubst$	String, String [,String]	String
Time$	String	Number
Today	-	Date
ToRTF$	String	String
UnitFromSCM	Number	Number
Upper$	String	String
Val	String	Number
Woy	Date, [Number]	Number
Year	Date	Number
Year$	Date	String

Abs

Purpose:

Returns the absolute value of a number. A negative value will be returned as positive and a positive value will remain unchanged.

Parameter:

Number

Return value:

> *Number*

Example:

```
Abs(-3) = 3
Abs(3.12) = 3.12
```

AddDays

Purpose:

> Adds the entered number of days to the date, or subtracts the number of days when a negative value is entered.

Parameter:

> *Date*
>
> *Number*

Return value:

> *Date*

AddHours

Purpose:

> Adds the entered number of hours to the date, or subtracts the number of hours when a negative value is entered.

Parameter:

> *Date*
>
> *Number*

Return value:

> *Date*

AddMinutes

Purpose:

> Adds the entered number of minutes to the date, or subtracts the number of minutes when a negative value is entered.

Parameter:

> *Date*
>
> *Number*

Return value:

> *Date*

AddMonths

Purpose:

Adds the entered number of months to the date, or subtracts the number of months when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddSeconds

Purpose:

Adds the entered number of seconds to the date, or subtracts the number of seconds when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddWeeks

Purpose:

Adds the entered number of weeks to the date, or subtracts the number of weeks when a negative value is entered.

Parameter:

Date

Number

Return value:

Date

AddYears

Purpose:

Adds the entered number of years to the date, or subtracts the number of years when a negative value is entered.

Parameter:

Date

Number

Return value:

> *Date*

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.
>
> During printing, a dialog will appear in which the needed information can be entered.
>
> The dialog allows the entered value to be carried over (button OK) or the function can be aborted (button Cancel).
>
> With the button "All", the entered value will be automatically used for all remaining open user entries. This is useful when the value remains constant over all records.

Parameter:

> *String* The first parameter contains the text that will appear in the dialog. Since this is text, this 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 opened once prior to printing, or if the dialog should be opened for each record. The following values are valid:
>
> False   The dialog appears once prior to printing.

124

True   The dialog appears for each record.

String (optional)
The third parameter contains the string that appears as the recommended value for the user input. Since this is text, this must be entered in quotation marks, for example "50,00 USD".

Number (optional)
The last parameter defines the number of digits that can be entered by the user. A value of 16, for example, allows the user to enter a maximum of 16 digits.

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 digits. Since the second parameter is TRUE, the dialog will open for each record to be printed.

ATrim$

Purpose:

Removes spaces from the beginning and end of a string.

Parameter:

String

Return value:

String

Example:

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

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

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

Barcode

Purpose:

This function converts a string to a barcode. This function can only be used in lists.

Parameter:

> ***String***

> ***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 7.6. Barcodes.

Return value:

> ***Barcode***

Example:

```
Barcode(Upper$(Name),"3of9")
```

Barcode$

Purpose:

> Returns the text contents of a barcode.

Parameter:

> ***Barcode***

Return value:

> ***String***

Example:

```
Barcode$(BC_3OF9)
```
  Result: "Item 4711"

BarcodeType$

Purpose:

> Returns the type of the barcode as a string.

Parameter:

> ***Barcode***

Return value:

> ***String***

Case$

Purpose:

> Converts a number, dependant upon the value, into a string. Assignment is made with a formatting string that contains the replacement string for the number values in ascending order.

Parameter:

> ***Number***   Number to be converted
>
> ***String***   Collection of linked character strings that are separated by a particular character. If a third parameter does not exist, this is the "|" character, otherwise the first character of this parameter. The n-th value of the character string will be copied to the return value string. If enough values do not exist, the character string will remain empty.
>
> ***String***   (optional) Separator for the formatting string

Return value:

> ***String***

Example:

> ```
> Case$(Page(),"0|I|II|III|IV|V|VI|VII|VIII|IX|X")
> ```
>
> Result: "III", if Page() = 3

CheckMod10

Purpose:

> Returns the modulo 10 check digit of the string.

Parameter:

> ***String***

Return value:

> ***Number***

Chr$

Purpose:

> Converts a number to a character. This character has the entered number as it's ASCII-Code.

Parameter:

> ***Number***

Return value:

> ***String***

Example:

`Chr$(64)` Result: "@"

ChrSubst$

Purpose:

Searches a character string for a string that is contained in the second parameter. Every occurence of this string will be replaced by the string defined in the third parameter. If no third parameter exists, the strings will be removed.

Parameter:

String

String

String (optional)

Return value:

String

Example

```
ChrSubst$("Otto", "Oo", "_")        Result: "_tt_"
ChrSubst$("Normalconsumer", "aeiou","??")  Result:
"N??rm??lc??ns??m??r"
ChrSubst$("Normalconsumer", "aeiou")  Result: "Nrmlcnsmr"
ChrSubst$("3.1415926535",".",",")   Result: "3,1415926535"
```

Cond

Purpose:

Allows conditions to be defined. The first parameter is a logical expression that will be evaluated as "True" of "False". If the first expression is "True", the second expression will be returned as the result. If the expression is "False", the third expression will be returned as the result. If no third expression is entered, the return value will assume the following standard values, dependent upon it's type:

2. Argument Type	Return value if expression = False
Boolean	False
String	"" (empty String)
Date	Julian Date value 0
Number	0
Drawing	"" (empty String)
Barcode	"" (empty String)

Parameter:

Boolean

All

All (optional)

Return value:

> *All*

Example:

> Imagine that your database contains items that are free samples and the items that you normally sell.
>
> Cond(PRICE=0,"Free Sample","Price:"+PRICE+" USD")

Contains

Purpose:

> Evaluates if a string contains another string (second parameter).

Parameter:

> *String*
>
> *String*

Return value:

> *Boolean*

Example:

> `Contains("Itemnumber: 12345", "1234")`   Result: True

Continued

Purpose:

> Indicates that a text or RTF-object had a page wrap. This means that the output is a result of the page wrap.

Parameter:

> *-*

Return value:

> *Boolean*

CStr$

Purpose:

> Formats a number with the assistance of a format character string. This is identical to the formatting information for the function printf() in the language C. The first parameter is a number of double precision, and the conversion operator can only assume the following values:
>
> 'f', 'g', 'G', 'e', 'E'.

Parameter:

> *Number*
>
> *String*

Return value:

> *String*

Example:

```
CStr(Pi,"%5.1f")          Result: " 3.1"
CStr(100*Pi,"nun: %g")    Result: "nun: 3.141593e+02"
```

Date

Purpose:

> Converts a string to a date.
>
> - If the string Contains a dot ".", it will be read in the "d.m.y" format (German).
>
> - If the string contains a diagonal slash "/", it will be read in the "m/d/y" format (US English).
>
> - If the string contains a dash "-", if will be read in the "y-m-d" format (ANSI).
>
> - If the input cannot be correctly interpreted, then the date represents a value that is larger than all other values, (1e100). The return value can be evaluated for correctness using "<JulianToDate(1e100).

Parameter:

> *String*

Return value:

> *Date*

Example:

```
Date("17.10.2002")
Date("10/17/2002")
Date("2002-10-17")
```

Tip:

> When one or two digit entries are made for the year, all values under 30 will be applied to the 21st century (20xx) and all values over 30 will be applied to the 20th century (19xx).

Date$

Purpose:

> Converts a date, using a format string, into an appropriately formatted string.
>
> Composition of the format string: this is a normal string into which placeholders can be embedded.

Place holder	Description
%d	Day (1..31)
%<n>d	Day to <n> digits
%0<n>d	Day to <n> digits, filled on left with '0's
%w	Weekday (1..7)
%<n>w	Weekday to <n> digits
%0<n>w	Weekday to <n> digits, filled on left with '0's
%m	Month (1..12)
%<n>m	Month to <n> digits
%0<n>m	Month to <n> digits, filled on left with '0's
%y	Year
%<n>y	Year, to <n> digits
%0<n>y	Year, to <n> digits, filled on left with '0's
%D	Weekday, written out
%M	Month, written out
"%e", "%<n>e"	Year in the local calendar (Japan: Emperor's year)
"%g", "%<n>g"	Era of the local calendar (Japan: Emperor's era)
"%g", "%1g"	Single letter, Latin letters
"%gg", "%2g"	Single letter, localized
"%ggg", "%3g"	Long name, localized
"%gggg", "%4g"	Long name, Latin letters
"%x"	Localized date, short form

As long as one of the above formats is used, the optional third parameter can be used to set the locale. If the second parameter contains a valid ISO-Country code, the third parameter can be used to set either the short "0" or long "1" format. See Chapter 7.5. List of the ISO 3166 Country Codes

Parameter:

> ***Date***   Value to be formatted
>
> ***String***   (optional)   Format description or country code
>
> ***String***   (optional)   Date format or country code

Return value:

> ***String***

Example:

```
Date$(Today(),"Date: %D, %d/%m/%y")
```
Result: "Date: Thursday, 7/11/2002"

```
Date$(Today(),"%2wthWeek; %D, %2d/%2m/%4y")
```
Result: "45th Week, Thursday, 7/11/2002"

```
Date$(Today(),"%D, %3d/%02m/%4y")
```
Result: "Thursday, 7/11/2002"

DateHMS

Purpose:

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

Parameter:

Number Hour

Number Minute

Number Second

Return value:

Date

DateInRange

Purpose:

Evaluates if the date falls within the entered time interval:

Minimum Date: JulianToDate(0)

Maximum Date: JulianToDate(1e100)

Parameter:

Date   Date to be evaluated

Date   Lower limit of the test interval

Date   Upper limit of the test interval

Return value:

Boolean

Example:

```
DateInRange(Date("2002.10.20"),Date("2002.2.29"),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())
```

DateYMD

Purpose:

Converts three numbers for day, month and year into a date.

Parameter:

Number Year

Number Month

Number Day

Return value:

Date

Example:

```
Date(2002, 11, 1)
```

Day

Purpose:

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

Parameter:

Date

Return value:

Number

Example:

```
Day$(Date("17.10.2002"))
```
   Result: 17

Day$

Purpose:

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

Parameter:

Date

Return value:

String

Example:

```
Day$(Date("17.10.2002"))
```
 Result: "17"

Dow

Purpose:

Returns the day of the week as a number(1..7), 1=Sunday, 2=Monday, ...

Parameter:

Date

Return value:

Number

Example:

`DoW(Today())` Result: 4

If the day for the current date is a Wednesday.

Dow$

Purpose:

Returns the day of the week as a string in accordance with the country settings, "Sunday", "Monday", ...

Parameter:

Date

Return value:

String

Example:

`DoW(Today())` Result: "Wednesday"

If the day for the current date is a Wednesday.

Drawing

Purpose:

Converts a string type file path into a drawing.

Parameter:

String

Return value:

Drawing

Drawing$

Purpose:

Converts a drawing into a string type file path.

Parameter:

Drawing

Return value:

> *String*

Empty

Purpose:

> Evaluates if a string is empty. If it is empty, "True" will be the return value, otherwise "False". Useful, for example, to determine if the field "ADDRESS" is empty, and if it is, in combination with the IF-THEN-ELSE condition cond(), either print the contents of the field "ADDRESS" or "POBOX".
>
> The third parameter allows the removal of leading and trailing spaces. If this is evaluated as "True", a string consisting of multiple spaces will be recognized as empty.

Parameter:

> *String*
>
> *Boolean* (optional)

Return value:

> *Boolean*

Example:

> `Empty("xyz")`   Result: False
>
> `Empty("")`   Result: True

Even

Purpose:

> Evaluates if a number is even. If the number is even, "True" will be returned, otherwise "False".

Parameter:

> *Number*

Return value:

> *Boolean*

Example:

> `"Pagenumber "+Cond(Even(Page()),"even","odd")`

Exp

Purpose:

> Calculates the exponential (e^x).

Parameter:

> *Number*

Return value:

> *Number*

Example:

> `Exp(3)`   Result: 20.08553692

Exp10

Purpose:

> Calculates 10 raised to the power of number (10^x).

Parameter:

> *Number*

Return value:

> *Number*

Example:

> `Exp10(3)`   Result: 1000

Frac

Purpose:

> Returns the decimal portion of a number

Parameter:

> *Number*

Return value:

> *Number*

Example:

> `Frac(Pi)`   Result: 0.1415926535

FStr$

Purpose:

> Formats a number with the help of 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 displayed.

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 strings assume the value 0
3	Removal of leading spaces and empty strings when value is 0

Parameter:

> ***Number***
> ***String***　　　Format string
> ***Number*** (optional)　Additional formatting

Return value:

> ***String***

Example:

```
FStr(3.142, "#")              Result: "3"
FStr(5003.1,"#,###.&&")       Result: "5.003,10"
FStr$(3.142,"#.###")          Result: "3,142"
FStr$(3.142,".######")        Result: "*******"
FStr$(3.142,"(#.###)")        Result: " 3,142 "
FStr$(-3.142,"(#.###)")       Result: "(3,142)"
FStr$(3.142,"+#.###")         Result: "+3,142"
FStr$(3.142,"-#.###")         Result: " 3,142"
FStr$(-3.142,"-#.###")        Result: "-3,142"
FStr$(3.142,"&&&.&&&")        Result: "003,142"
FStr$(3.142,"***.***")        Result: "**3,142"
FStr$(3.142,"$$$.$$$")        Result: "$$3,142"
FStr$(3.142,"###.***")        Result: "  3,142"
FStr$(5003.1,"#,###.&&")      Result: "5.003,10"
FStr$(3.142,"#####")          Result: "    3"
```

Hour

Purpose:

Determines the hour of the entered date and returns it in number format. If the parameter is not used, the hour of the print time will be returned.

Parameter:

Date (optional)

Return value:

Number

Example:

A condition can evaluate if the current hour has the value "10". The value of the current hour must be determined and then compared to the value "10".

```
Hour()=10
```

Hyperlink$

Purpose:

The function Hyperlink$ creates a hyperlink text that can be inserted using an export module.

When available, the hyperlink will only be embedded if the third parameter returns a result of "True" (=default).

Parameter:

String	Text
String	Hyperlink
Boolean	(Optional) Embed

Return value:

String

Example:

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

Tip:

To optionally use a hyperlink only on a HTML, you need to use the List & Label variable LL.OutputDevice:

```
Hyperlink$("combit","http://www.combit.net",LL.OutputDevice="HT
ML")
```

If an object text contains the character string:

```
<!--begin:hyperlink="target"-->"Display text"<!--end:hyperlink-
->
```

138

then a hyperlink will be automatically created on the target. The hyperlink function automatically creates a string with the correct syntax.

If

see Cond

Int

Purpose:

Returns the whole number portion of a number.

Parameter:

Number

Return value:

Number

Example

`Int(3,1)`   Result: 3

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(2452568)`   Result: 2002.10.21

LastPage

Purpose:

Returns if the current page is also the last page.

Note: This function can only be used in the footer lines of tables, or in objects linked with tables! In all other cases, the result of Lastpage() is always False.

Parameter:

-

Return value:

Boolean

Example:

```
Cond(Lastpage(),"Total sum","Subtotal")
```

Left$

Purpose:

Shortens a string from the right so that only the number of characters are visible that correspond to the number entered in the parameter. If the original string matches the parameter, it will not be changed.

Parameter:

String

Number

Return value:

String

Example:

Assume that you have a customer database that contains, among others, the field NAME. You wish to find all entries that begin with the letter C.

The expression Left$(NAME, 1) returns the first letter of the string NAME.

```
Left$("combit", 2)
```
  Result: "co"

Len

Purpose:

Returns the number of characters in a string.

Parameter:

String

Return value:

Number

Example:

```
Len("1234"+"12")
```
  Result: 6

Locale$

Purpose:

Returns information about the country settings, for example currency, decimals, separators, language and country code. The code for the appropriate country is entered in the second parameter, if no second parameter is used the default country settings will be used.

Parameter:

> ***Number***   Index of Locale Entry
>
> ***String***   (optional)   Country code

Return value:

> ***String***

Example:

> `Locale$(42,"USA")`   Result: "Monday"
>
> See also Chapter 7.5. List of the ISO 3166 Country Codes.

LocCurr$

Parameter:

> ***Number***   Value to be formatted
>
> ***String*** (optional)   Country code

Purpose:

> Returns a string with the valid currency format without the currency symbol for the entered country.

Example:

> `LocCurr$(123,"USA")` Result: "123.00"
>
> See also Chapter 7.5. List of the ISO 3166 Country Codes.

LocCurrL$

Parameter:

> ***Number***   Value to be formatted
>
> ***String*** (optional)   Country code

Purpose:

> Returns a string with the valid currency format and currency symbol for the entered country.

Example:

> `LocCurr$(123,"USA") -> "$123.00 "`
> See also Chapter 7.5. List of the ISO 3166 Country Codes.

LocDate$

Parameter:

> ***Date***   Value to be formatted
>
> ***String*** (optional)   Country code
>
> ***Number*** (optional)   Format

Purpose:

Returns a string with the valid date format for the entered country.

Example

`LocDate$(Date("17.11.2002"),"USA")` Result: "11/17/2002"

See also Chapter 7.5. List of the ISO 3166 Country Codes.

LocNumber$

Parameter:

Number   Value to be formatted

String (optional)   Country code

Purpose:

Returns a string with the valid number format for the entered country.

Example:

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

See also Chapter 7.5. List of the ISO 3166 Country Codes.

LocTime$

Parameter:

Date   Value to be formatted

String (optional)   Country code

Number (optional)   Format

Purpose:

Returns a string with the valid time format for the entered country.

Example:

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

See also Chapter 7.5. List of the ISO 3166 Country Codes.

Log

Purpose:

Returns the value of the natural logarithm (ln(x)).

Parameter:

Number

Return value:

Number

Example:

    Log(Exp(1))   Result: 1

Log10

Purpose:

Returns the value of the logarithm to base 10 (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:

Compares two values of number or date type, and returns the larger value.

Parameter:

> ***Number*** or ***Date***
>
> ***Number*** or ***Date***

Return value:

> ***Number*** or ***Date***

Mid$

Purpose:

> The resulting string will be extracted from the original. 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:

> Compares two values of number or date type and returns the smaller value.

Parameter:

> ***Number*** or ***Date***
>
> ***Number*** or ***Date***

Return value:

> ***Number*** or ***Date***

Minute

Determines the minute of the entered date, and returns the result as a number. If the parameter is not used, the minute of the time of printing will be returned.

Parameter:

Date (optional)

Return value:

Number

Example:

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

```
Minute()=10
```

Month

Purpose:

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

Parameter:

Date

Return value:

Number

Example:

`Month(Date("2002.10.17"))` Result: 10

Month$

Purpose:

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

Parameter:

Date

Return value:

String

Example:

`Month$(Date("2002.10.17"))` Result: "10"

Now

Purpose:

Returns the current date and time.

Parameter:

> *-*

Return value:

> *Date*

NumInRange

Purpose:

> Evaluates if a number falls within the desired range.

Parameter:

> *Number*
>
> *Number*   Upper limit
>
> *Number*   Lower limit

Return value:

> *Boolean*

Example:

```
NumInRange(Page(),1,10)
```

> Result: True, if the page number is between 1 and 10.

Odd

Purpose:

> Evaluates if a number is odd. If the number is odd "True" will be returned, otherwise "False".

Parameter:

> *Number*

Return value:

> *Boolean*

Example:

```
"Pagenumber "+Cond(Odd(Page()),"odd","even")
```

Ord

Purpose:

> Returns the ASCII value of the first character.

Parameter:

> *String*

Return value:

> *Number*

Example:

    `Ord("Combit")`   Result: 67

Page

Purpose:

Returns the current page number.

Parameter:

-

Return value:

Number

Example:

    `Case(Odd(Page()),"Even","Odd")+" Pagenumber"`

Pow

Purpose:

Corresponds to the function (Base) ^ (Exponent).

Parameter:

Number Base

Number Exponent

Return value:

Number

Example:

    `Pow(2,3)`   Result: 8

ProjectPath$

Purpose:

Returns the path of the current project.

Parameter:

-

Return value:

String

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:

>Shortens a string from the left so that the number of characters entered in the parameter remain. If the original string is already small enough, the string will not be changed.

Parameter:

>*String*
>
>*Number*

Return value:

>*String*

Example:

>`Right$("Normalconsumer", 8)`   Result: "consumer"

Round

Purpose:

Rounds a value to the entered number of decimal places. Default is 0.

Parameter:

Number

Number (optional)

Return value:

Number

Example:

Round(3.1454,2)   Result: 3,15

Round(3.1454)   Result: 3

RTrim$

Purpose:

Removes spaces from the end of a string.

Parameter:

String

Return value:

String

Example:

RTrim$("Otto   ")   Result: "Otto"

Second

Determines the second of the entered date and returns the result as a number. If the parameter is not used, the second of the print time will be returned.

Parameter:

Date (optional)

Return value:

Number

Sqrt

Purpose:

Returns the square root of a number.

Parameter:

Number

Return value:

> *Number*

Example:

> `Sqrt(4)`   Result: 2

Str$

Purpose:

> Converts a number to a string. The number will be formatted with 6 decimal places that may be rounded. The length is variable.

> The second parameter defines the length of the string. 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).

> The third parameter defines the number of decimal places. If the number is positive, the number will be displayed as a floating-point number, if negative in scientific format.

Parameter:

> *Number*
> *Number*
> *Number* (optional)

Return value:

> *String*

Example:

```
Str$(Pi)          Result: "3.141593"
Str$(Page())      Result: "5.000000"
Str$(Pi,3)         Result: "3.1"
Str$(Page(),10)   Result: "  5.000000"
Str$(Page(),-10)  Result: "5.000000  "
Str$(Pi,3,0)      Result: "  3"
Str$(-Pi,12,-3)   Result: "  -3.141e+00"
```

StrPos

Purpose:

> Returns the position of the nth appearance of a search string. The third parameter determines which appearance will be returned. Default is 1.

> The first character in the string has the position 0.

> -1 as return value signifies the search string no longer appears.

Parameter:

> *String*
>
> *String*   Search string
>
> *Number* (optional)

Return value:

> *Number*

Example:

```
StrPos("Normalconsumer","or")        Result: 1
StrPos("Normalconsumer","r")         Result: 2
StrPos("Normalconsumer","r",1)       Result: 2
StrPos("Normalconsumer","r",2)       Result: 13
```

StrRPos

Purpose:

> Returns the position of a search string within a string. This is a backwards search. The third parameter, determines which appearance will be returned. Default is 1.
>
> The first character in the string has the position 0.
>
> -1 as return value signifies the search string no longer appears.

Parameter:

> *String*
>
> *String*   Search string
>
> *Number*   (optional)

Return value:

> *Number*

Example:

```
StrRPos("Normalconsumer","or")       Result: 1
StrRPos("Normalconsumer","r")        Result: 13
StrRPos("Normalconsumer","r",1)      Result: 13
StrRPos("Normalconsumer","r",2)      Result: 2
```

StrSubst$

Purpose:

> Searches a string for the appearance of a search string and replaces it with the string contained in the third parameter (replacement string). If no third parameter is used, the string located using the search string will be deleted.

Parameter:

> *String*

> *String*   Search string

> *String* (optional)   Replacement string

Return value:

> *String*

Example:

> Assume that you want to print address labels that contain the company name. You do not have much space available on the label and cannot afford to completely print long company names, for example, "Forrer Construction, Incorporated".

> With the expression `StrSubst$(COMPANY,"Incorporated","Inc.")` every appearance of "Incorporated" in the COMPANY field will be replaced with "Inc."

Time$

Purpose:

> Returns the current time in string format.

> The following formats are available:

Placeholder	Description
%h	Hours in 24 hour format
%H	Hours in 12 hour format
%m	Minutes
%s	Seconds
%P	Display the part of day (A.M. / P.M.)
%p	Display the part of day (a.m. / p.m.)

Parameter:

> *String*

Return value:

> *String*

Example:

> `Time$("%02h:%02m:%02s")    Result: "18:30:45"`

Today

Purpose:

> Returns the current date.

Parameter:

> **-**

Return value:

> *Date*

Example:

```
Date$(Today(),"%D, %m.%d.%4y")
```

Result: "Friday, 11/8/2002"

ToRTF$

Purpose:

Returns a string in RTF-format. This is necessary because some strings may contain one of the specially defined RTF-format symbols. ('\', '{' or '}').

Parameter:

> *String*

Return value:

> *String*

Example:

> If, for example, the field PRODCODE could contain one of the characters, then the text should be inserted in the following way:

```
"<<SALUTATION>> <<NAME>>, You have received our product
<<PRODUCT>>, Code <<ToRTF$(PRODCODE)>>..."
```

UnitFromSCM

Purpose:

> Converts a SCM-Unit (1/1000 mm) to the print unit (inch/mm). Important for the definitions of property values independent of the selected print unit.

Parameter:

> *Number*

Return value:

> *Number*

Example:

```
Cond(Page()=1,UnitFromSCM(100000),UnitFromSCM(20000))
```

Result: 10cm for Page 1, 20 cm for the other pages.

Upper$

Purpose:

Converts the characters of a string to capital letters.

Parameter:

String

Return value:

String

Example:

`Upper$("Otto")`   Result: "OTTO"

Val

Purpose:

The string is interpreted and returned as a number. If an error occurs, the return value is 0. The decimal sign must always be entered as ".".

Parameter:

String

Return value:

Number

Example:

`Val("3.141")`   Result: 3.141
`Val("3,141")`   Result: 3
`Val("3.141e2")`   Result: 314.2
`Val(ChrSubst$("3,141", ",", "."))`   Result: 3.141

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: 2002
Year$(Date("1.1.2004"))  Result: 2004
```

Year$

Purpose:

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

Parameter:

Date

Return value:

String

Example:

```
Year$(Today())            Result: "2002"
Year$(Date("1.1.2004"))   Result: "2004"
```

7.5. List of the ISO 3166 Country Codes

The following country codes can be used with, among others, the Loc.. functions. (CountryCode: Prefix):

Code	Prefix	Code	Prefix	Code	Prefix	Code	Prefix	Code	Prefix
ALB	355	BRA	55		3		4	LUX	352
ARG	54	CAN	2	ESP	34	HRV	385	LVA	371
AUS	61	CAR	1	EST	372	HUN	36	MEX	52
AUT	43	CHE	41	FIN	358	IDN	62	NIC	505
BEL	32	CHL	56	FRAU	33	IRL	353	NLD	31
BGR	359	COL	57	FRO	298	ISL	354	NOR	47
BLR	375	CRI	506	GBR	44	ITA	39	NZL	64
BLZ	501	CZE	42	GRC	30	JAM	1	PAN	507
BOL	591	DEU	49	GTM	502	LIE	41	PER	51
		DNK	45	HND	50	LTU	370	POL	48
		DOM	1						
		ECU	59						

PRI	1
PRT	351
PRY	595
ROM	40
RUS	7
SLV	503
SPB	381
SVK	42
SVN	386
SWE	46
TTO	1
TUR	90
UKR	380
URY	598
USA	1
VEN	58
ZAF	27

7.6. Barcodes

List & Label supports a wide range of barcode formats. Normally, no special printers, fonts, etc. are necessary, the barcodes will be printed directly from List & Label.

7.6.1. List of Available Barcodes

EAN13, JAN13

The EAN13-(International Article Numbering)-Code is a very common barcode. The number string used with EAN13 must conform to one of the following formats:

```
"cc|nnnnn|aaaaa"   (normal EAN13)
"cc|nnnnn|aaaaa|xx"   (EAN13 for periodicals, "ISSN")
"cc|nnnnn|aaaaa|xxxxx"   (EAN13 for books, "Bookland")
"ppp|nnnn|aaaaa"   (normal EAN13)
"ppp|nnnn|aaaaa|xx"   (EAN13 for periodicals, "ISSN")
"ppp|nnnn|aaaaa|xxxxx"   (EAN13 for books, "Bookland")
```

with

	Description	Value Range
Cc	Country code	c='0'..'9'
Ppp	Product code	
Nnnnn	Company code	n='0'..'9'
aaaaa	Article code	a='0'..'9'
'|'	= Character code chr(124)	
xx, xxxxx	Supplemental code	x=´0´..´9´

The check digit will be calculated and attached automatically. Each character is 7 bar-widths wide, a code should have a minimum width of (12*7+11)*0.3 mm = 2.85 cm.

Ideal size (bar symbol) nominal size SC2:

Width = 31.4 mm
Height = 24.5 mm

Minimum offset that should be kept free around the symbol:

left: 3.6 mm
top: 0.3 mm
right: 2.3 mm
bottom: 0.0 mm

The text may partially exceed this area.

EAN14, UCC14

These number codes require a high print quality.
Permitted characters: '0'..'9'. The code must be 14 characters long.

EAN8, JAN8

The number string in EAN8 must conform to the following format:

"nn|nnnnn"
with

- n   = '0'..'9'
- | = Character code chr(124)

Each character is also 7 bar widths wide, a code should then have a minimum width of (8*7+11)*0.3 mm=2.01 cm.

Ideal size (bar symbol)   nominal size SC2:

- dx :   22.1 mm
- dy: 19.9 mm

Minimum offset that should be kept free around the symbol:

- left: 2.3 mm
- top: 0.3 mm
- right: 2.3 mm
- bottom: 0.0 mm (if printed, otherwise 0.3 mm)

The text may partially exceed this area.

UPC-A

The UPC-A-Code (Universal Product Code) is common in the USA. The code must conform to the following format:

"c|nnnnn|aaaaa"
with

- c   = Number system
- nnnnn   = Company code
- aaaaa   = Article code
- | = Character code chr(124)

The check digit will be calculated and attached automatically. Each character is also 7 bar widths wide, a code should then have a minimum width of (13*7+6)*0.3 mm=2.88 cm.

UPC-E

The UPC-E-Code (Universal Product Code, short version) is common in the USA. The code must conform to the following format:

"c|nnnnn"

with

- c   = Number system
- | = Character code chr(124)
- nnnnn   = Code, interpretation dependent upon the last digit

The check digit will be calculated and attached automatically. Each character is also 7 bar widths wide, a code should then have a minimum width of (13*7+6)*0.3 mm=2.88 cm.

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

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

Permitted characters: ' ','$','/','%','*','+','-','0'..'9','A'..'Z'

The expanded code can be activated by a combination of the standard code: for example: '+A' -> 'a'. Every character is 16 bar widths wide, a text has (16*number of characters −1) bars.

2-of-5 Industrial

A number code of low information density.
Permitted characters: '0'..'9'
A Code is (14*number of characters+18) bar widths wide.

2-of-5 Interleaved (ITF)

A number code of high information density, requires high print quality
Permitted characters: '0'..'9'. The number of characters must be even.
A code is (9*number of characters+9) bar widths wide.

2-of-5 Matrix

A number code of high information density, requires high print quality.
Permitted characters: '0'..'9'.
A Code is (10* number of characters +18) bar widths wide.

2-of-5 Datalogic

A number code of high information density, requires high print quality.
Permitted characters: '0'..'9'.
A code is (10* number of characters +11) bar widths wide.

Postnet

This is a barcode used by the United States Postal Service. It is available in three variants. In reference to placement and offset from other objects, please refer to the specifications in the appropriate literature.

5-digit: "nnnnn"
10-digit: "nnnnn-nnnn"
12-digit: "nnnnn-nnnnnn"
Minimum size: 1.245" * 4/16" = 31.6 mm * 6.35 mm (10-digit).

Minimum bar spacing 1/24" = 1.058 mm.

The error-correction digit will be automatically amended.

This barcode will be automatically printed in the correct size.

FIM Barcode

Minimum size: 1/2" * 5/8" = 12.7 mm * 15.87 mm.

Transfer values: "A", "B" or "C".

The FIM-Barcode is always printed to the size required by the United States Postal Service. It can expand beyond the object frame.

Codabar (2)

The Codabar-Code is a numerical code with 6 special characters. The information density is low. The code must conform to the following format:

"fnnnnf"

with

- f   = Frame code ('A', 'B', 'C', or 'D')
- nnnnn   = arbitrary quantity of numbers or special characters ('0'..'9', '-', '$', ':', '/', '.', '+')

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

EAN128

This code can display all printable ASCII-characters, as well as non-explicit characters, umlauts and 'ß'. The text can have any appearance. The width is not easy to display because certain characters have different widths.

Special characters   must be replaced with:
Character   Replacement [using chr(...), chr$(...)]
NUL   chr$(255)
FNC1   chr$(254)
FNC2   chr$(253)
FNC3   chr$(252)
FNC4   chr$(251)

Code128

This code is basically identical to EAN128, with the exception that the first character (FNC1 ... FNC4) is defined by the user.

The explanations with EAN128 apply here as well.

DP-Identcode

A number code of high information density, requires high print quality. Permitted characters: '0'..'9'.

The code must conform to the following format:

```
"nn.nnnnn.nnnn"
"nn.nnnn.nnnnn"
"nn.nnn.nnnnnn"
```
A Code is (9*Number+9) bar widths wide.

Width: 32,0mm - 58,5mm (Clear zone right and left: minimum 5mm). height: 25mm.

The check digit is calculated automatically; Relation: 4:9; Special code of 2 of 5 IL.

DP-Leitcode

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

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

The code must conform to the following format:

```
"nnnnn.nnn.nnn.nn"
```
A Code is (9*Number+9) bar widths wide.

Width: 37,25 mm - 67,5 mm (Clear zone right and left: minimum 5mm). Height: 25mm.

The check digit is calculated automatically; Relation: 4:9; Special code of 2 of 5 IL.

German Parcel

A number code of low information density.

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

A Code is (14*Number+18) bar widths wide.

Relation: 1:2

Code93 (simple and extended)

Code93 is an extension of Code39, but has the advantage that it is somewhat smaller. It covers the complete 128 bytes of the ASCII character set, including zero. This must be transferred as chr$(255).

It contains two check digits that are automatically generated by List & Label.

The characters consist of 9 bar widths, that each have 3 bars and 3 spaces. There are two options for the extended code:

- transfer of the shift character from the host program as

```
$   chr$(254)
%   chr$(253)
/   chr$(252)
+   chr$(251)
```
- transfer of the desired character, L&L adds the appropriate shift character.

MSI

MSI is a binary barcode in which every character consists of 8 bars.

The character set is limited to '0'..'9' and is suitable only for the display of numbers. Commonly used in libraries.

CODE11

Character set: '0'..'9','-'.

Code 11 has, depending on the length, 1 or 2 check digits. List & Label calculates only 1 instead of 2 check digits if the length of the text is a maximum of 10 characters.

Maxicode and PDF417

(Application dependent)

Character set: All available and also non-printable characters. To use non-printable characters in barcode text, tags in the form {binary:xx} can be used in the barcode text, with xx standing for any sequence of two digit hexadecimal numbers. This is especially important when Maxicodes are to be created to UPS specifications. The required special characters can be entered in this way.