Version

6.0

BAR-ONE®

User’s Guide

English

The information in this documentation is not contractual in nature. It is subject to modification without notice.

Table of Contents

About this Manual

Welcome!

Congratulations, you have just purchased the world’s leading automatic identification software package!

This label design software allows you to create and edit complex labels—fast—and is capable of integrating all the electronic data from across your company, guaranteeing the full integrity of all your systems.

The application meets all your automatic identification needs, and fits perfectly into your production process.

On today’s market, it is quite simply the **MOST USER-FRIENDLY AND INTUITIVE SOLUTION** for any company involved in complex barcode data entry, collection and editing in real time.

Documents supplied

Complete documentation is provided to help you make full use of all the resources offered by your label design software.

The *Quick Start Guide* takes you through product installation and provides an introduction to the basic functions of the software interface.

The *User’s Guide* presents the basic concepts one by one, helping you to get the very most from the software. A wealth of examples enables you to put the ideas covered straight into practice.

The documentation has been designed for use in conjunction with the integrated Online Help.

The User's Guide

The purpose of this manual is to get you off to a quick start with the software by familiarizing you with its most common functions, then covering the more complex functions by means of examples of label creation.

The *User's Guide* is divided into six main sections :

- Discovering the interface
- Setting up the work environment
- Creating a document using fixed objects
- Creating a document using variable objects
- Printing
- Security and User Manager

The four annexes provide detailed information and examples on some of the more complex functions:

- formulas
- ODBC and importing data
- counters
- ActiveX

This manual outlines the surest and most effective way of carrying out specific tasks. For more detail on the different functions, step-by-step procedures and reference information, use the Online Help.

This manual has been designed to provide you with the most effective help possible, whether you are a beginner or an experienced user. The exercises are presented in increasing levels of complexity, allowing you to save time by moving directly to those at your level.

Remarque

Beginners are recommended to follow the exercises through step by step.
The examples provide an general approach to the many functions. However, given the richness of the software, they are not intended to be exhaustive.
The *User's Guide* is complimented by the Online Help, and is by no means intended as a substitute for it.

Typographical conventions

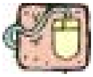
This manual distinguishes between different types of information by using the following conventions:

- terms taken from the interface itself, such as commands, appear in **bold**;

- keys appear in small caps, as in the following example: ”Press the SHIFT key”;
- numbered lists mean there is a procedure to be followed;
- when the conjunction ”or” appears next to a paragraph it means there is the choice of another procedure for carrying out a given task;
- When a menu command contains submenus, the menu name followed by the command to select appears in bold. Thus, ”Choose **File Open**” means choose the **File** menu, then the **Open** command.

This symbol highlights important information about how a particular command or procedure works.

Following this symbol you will find hints and tips for optimizing tasks, speeding up commands, and so on.



This symbol indicates an introductory exercise for you to work through.

About your product

Some of the functions described in this manual may not be available in your product.

For the complete list of the specific features and functions available in your software, please refer to the specifications sheet provided with the product.

Discovering the Interface

1

Launching the program

1 Choose **Start Programs**.

2 Then select the name you gave to the program group when you installed it (by default, it is the name of the application).

The main window appears on the screen.

Description of the main window

This section presents a general overview of the main elements that make up the interface, as they appear in the main window at the beginning of a work session.

Menu bar

The menu bar comprises eight drop-down menus: **File, Edit, View**, **Object**, **Merge**, **Tools**, **Window** and **Help**.

▶ **To open a menu**:

3 Select it with the left mouse button.

4 Then choose the required command.

To access commands via the keyboard, use the shortcut keys. Press ALT plus the key corresponding to the letter underlined in the menu name, then the key corresponding to the letter underlined in the command name.

Workspace

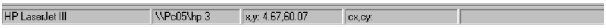
The workspace is the name given to the entire middle section of the window, located between the graduated rulers and the scroll bars. It includes the frame that sets the physical limits of the document in which you will place the objects to be printed, as well as a non-printable area where you can insert comments or objects that you do not wish to print.

Document

When you open a new session, the program displays a frame aligned with the top left-hand corner of the window. This frame represents the physical limits of the document to be printed. The objects that make up your document should be placed inside this frame.

Status bar

Located on the lower edge of the window, the status bar displays the name of the selected printer, the communication port it is connected to, the x and y coordinates of the mouse pointer and the cx and cy dimensions of the object selected.



HP LaserJet III | \\Pc05\hp 3 | x,y: 4.67,60.07 | cx,cy:

Figure 1 The status bar

Standard toolbar

These tools allow you to carry out routine tasks more quickly than by using the menus.

▶ **To select a tool:**

- Click on the button corresponding to the tool.

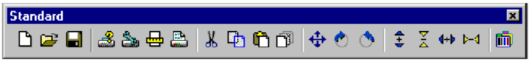


Figure 2 The **Standard** toolbar

Text formatting toolbar

These tools allow you to modify text and/or paragraph formatting, i.e. change the font, character size, style options, text wrap, and so on.

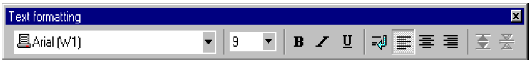


Figure 3 The **Text formatting** toolbar

▶ To change the formatting:

1 Select an object in your document.

2 Click on the required options.

All these options can also be accessed in the **Text** dialog box by the command **Object Text...**

Object formatting toolbar

These tools allow you to change the formatting of a selected object, i.e. place it in the background or foreground, designate it printable or non-printable, lock or unlock it, modify line thickness, or change the position of the barcode human readable.

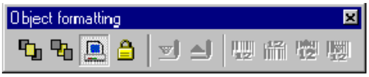


Figure 4 The **Object formatting** toolbar

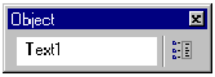
▶ To change the formatting:

1 Select an object in your document.

2 Click on the required options.

The Object toolbar

The **Object** toolbar displays the name of the selected object and allows you to access its properties.



Text1

Figure 5 The **Object** toolbar

▶ To access the object’s properties:

- Click on

View toolbar

The **View** toolbar offers you a number of display options.

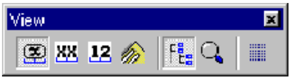


Figure 6 The **View** toolbar

▶ **To change the view mode:**

- Click on the required view mode:

Button	Name of tool	Use
	Name	displaying variable names.
XX	**Size**	displaying variables sizes. The space occupied by the content of the variable is represented by a series of Xs.
12	**Contents**	displaying variable values.
	Form	displaying the **Form**.
	Document Browser	displaying the **Document Browser**.
	Zoom	allows you to enlarge or reduce object detail.
	Grid	displaying the grid on the screen.

All these options can also be accessed via the **View** menu.

Creation toolbar

The **Creation** toolbar comprises a number of buttons, each one representing a tool for creating and manipulating objects.

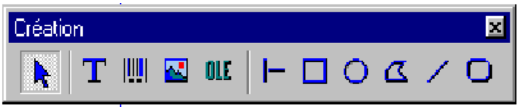


Figure 7 The **Creation** toolbar

Button	Name of tool	Use
↖	**Selection tool**	selecting and manipulating objects of all types
T	**Text generation tool**	creating fixed and variable text
‖‖	**Barcode generation tool**	creating fixed and variable barcodes
🖼	**Image import tool**	inserting images
OLE	**External object insertion tool**	inserting objects external to the application
⊢	**Line drawing tool**	creating horizontal or vertical lines
□	**Rectangle drawing tool**	creating rectangles or squares
○	**Circle or ellipse drawing tool**	creating circular shapes
⬠	**Polygon drawing tool**	creating free shapes made up of lines
/	**Oblique line drawing tool**	creating diagonals
▢	**Rounded rectangle drawing tool**	creating rectangles with rounded corners

Document Browser

The **Document Browser** contains two tabs: **Data Sources** and **Objects.**The + and - symbols allow you to move up and down the tree and display the list of variables associated with each data source or the objects available for each category.

The **Data Sources** tab displays the data sources and the number of variables associated. It allows you to insert variable objects simply by dragging and dropping them into your document. You can also add, delete and modify variable properties using the context menu.

The **Objects** tab displays the number of objects created by type, and their properties such as position, height and width. You can select an object and modify its position using this tab.

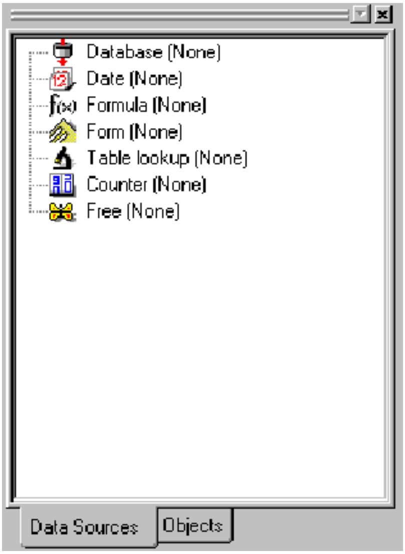


Figure 8 The **Document Browser**

▶ About data sources

A data source contains a list of variables fed by data of the same origin.

When you select a data source, you are indicating what type of data is required and where it will come from. Data can be of internal origin (Formula, Date, Counter) or external to the application (Databases, direct keyboard input).

The different data sources available in the **Document Browser Data Sources** tab are the following: **Database**, **Table lookup**, **Date**, **Formula**, **Form**, **Counter**, and **Free**.

Graduated rulers

These display the position of the mouse pointer and allow you to place the various objects included in your document with total precision. They can be configured in inches or millimeters using the **Tools Options... Display** command.

- A double-click on the left half of the horizontal ruler reduces the display scale.
- A double-click on the right half of the horizontal ruler increases the display scale.
- A double-click on the vertical ruler displays the grid selection box.

Using the right mouse button, you can display the **Zoom** factor context menu by clicking on the horizontal ruler, or the **Orientation** context menu by clicking on the vertical ruler.

Rotation button

This button allows you to rotate the document on the screen through 90°, 180°, 270° or 360°. Screen rotation is merely a display feature, and has no effect on printing.

▶ **To rotate the document:**

- Click on the rotation button with the left mouse button to rotate the document through 90° to the left.
- Click on the rotation button with the right mouse button to rotate the button through 90° to the right.

You can obtain the same document rotation effect by choosing the **View Orientation** command, or by clicking on the vertical ruler with the right mouse button to display the context menu.

Color palette The color palette, displayed along the lower edge of the workspace, allows you to modify the color of all your objects (text, barcodes, shapes and black-and-white images).

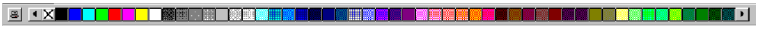


Figure 9 The **Color palette**

▶ **To change the color of an object:**

1 Select an object in your document.

2 Click on a color in the palette with the left mouse button to apply text, line or outline color.
OR
Click on a color in the palette with the right mouse button to apply background color.

Note

The button shown below allows object background color to be removed.

The Online Help provides full information on choosing the color palette, customizing it, and so on.

Setting Up the Work Environment

2

Before starting to create a document, you first need to set up your work environment.

The first step consists of selecting the printer on which you wish to print your labels. Not all printers share the same characteristics and, as a result, different page setup options, for example, will be available according to the model of printer chosen.

The second step involves choosing your work environment options, i.e. interface language, unit of measurement, and so on. We will review all the options available later in this chapter.

The final step consists of the document page setup, i.e. defining the dimensions of the document, orientation, margins, and so on.

Choice of printer

Your label design software allows you to print on a wide range of printers, the drivers for which are supplied with the software. You can also print on any Windows™ printer installed locally on your machine or on your company's network.

You must select the printer before you begin creating your document as the choice of printer will determine page setup options, the use of different character fonts, and so on.

Selecting the printer

The printer selection dialog box can be accessed using the **File Select printer...** command, by the 🖨 button or by pressing F5.

It can also be accessed via the **Printer...** button in the **Print** dialog box (F6 key).

▶ **To select a printer:**

1 Click on the 🖨 button.
OR
Choose **File Select printer...**

You can also press the F5 key.

2 Select the printer from the list, then click on OK.

Note

The printer drivers provided by the application are indicated by the following icon:

🖨

Adding a printer

The **Add printer** dialog box is accessed using the **File Select printer...** command, then by clicking on **Add...**

This dialog box allows you to add the printer drivers supplied with the software.

▶ **To add a printer:**

1 Choose **File Select printer...**

You can also click on the 🖨 button or press the F5 key.

2 Click on **Add...**

3 In the **Model** list, select the make then model of printer.

4 Select the communication port (serial, parallel, file or network), then click on OK (see **Printing**).

Click on the + sign next to the make of printer to display the different models.

Customizing the work environment

Your software offers a number of options for customizing your work environment.

Modifying display and work environment options

Options include the choice of language interface, unit of measurement, dialog box display font, working folders, and currency conversion rate.

These choices are made in the **Options** dialog box.

- To access this dialog box, choose **Tools Options...**

▶ **To change the interface language:**

1 Choose **Tools Options...**

2 Click on the **Display** tab.

3 Select the required language, then click on OK.

Note

Once you click on OK, all the menus, commands and screen messages will appear in the selected language (except the Online Help and the Windows dialog boxes such as **Open**, **Save**, and so on).

▶ To choose a unit of measurement:

1 Choose **Tools Options...**

2 Click on the **Display** tab.

3 Select the unit, then click on OK.

▶ To change the dialog box display font:

1 Choose **Tools Options...**

2 Click on the **Display** tab.

3 In the **Display font** area, click on ... .

4 Select the font and size, then click on OK.

▶ To modify the grid scale:

1 Choose **Tools Options...**

2 Click on the **Grid** tab.

3 Select the required grid scale.

4 Select the **Display** option, then click on OK.

More advanced!
Double-click on the vertical ruler to access the **Grid** tab directly.

▶ To define a conversion rate:

1 Choose **Tools Options...**

2 Click on the **Others** tab.

3 Enter the conversion rate in the box, then click on OK.

Note

The default conversion rate is between the French Franc and the Euro. To change to a different currency system, simply enter the corresponding rate.
The conversion rate defined in the box is the one used in the **eurotocurrency** and **currencytoeuro** formulas.

▶ To define a working folder:

1 Choose **Tools Options...**

2 Click on the **Default folders** tab.

3 In the **File type** list, select the type of file for which you wish to define a new default folder.

4 Enter the new access path or click on **Browse...**

5 Enter or select the access path for the new working folder.

6 Repeat this procedure for each file type as required, then click on OK.

Document Page Setup

Document page setup consists of specifying the page size, orientation, number of documents per row and per column, document size, margins, and distance between documents. You can also create customized stocks and save them for future use.

Document stocks are created and modified using the **Page setup** dialog box.

- To access the **Page setup** dialog box, click on the button
OR
Choose **File Page setup...**

Note

Before creating a document stock, you must first choose the model of printer that will be used to print the document (see Chapter 2 - 10). Page format is defined in the **Page setup** dialog box, which also updates the printer settings. The software does not use the page settings defined in the **Printer setup** dialog box.

Saving a customized stock

Once you have finished defining your stock, you can then save it for future use. Stocks are managed by saving them under different categories, in much the same way as the folders in Windows™ Explorer. Saving stocks in this way makes them much easier to relocate.

▶ **To save a customized stock:**

1 Choose **File Page setup...**
OR
Click on the button.

2 Set up the format.

3 In the **Stock** tab, select a type from the **Type** list, or enter a new one.

4 Select a stock name from the **Name** list, or enter a new one.

5 Click on **Save**.

Note

If you are not happy with the settings you have made, just click on **Remove**. The program closes the **Page setup** dialog box without saving the changes.
On the other hand, the predefined stocks supplied with the application cannot be modified.

Designing a CD sleeve: page setup and creating a customized stock.

1 Create a new document and call it FRONT.

2 Select the printer.

3 Choose **Tools Options...**, click on the **Display** tab and select **Millimeters** as the unit of measurement. Click on OK.

4 Choose **File Page setup...**
OR
Click on the button.

Note

The printer used in this example is a Windows™ printer. If you wish to use a **thermal** printer, it may be that error messages will appear during page setup. You will thus need to modify the values suggested in the example because, as we have already seen, the choice of printer affects certain settings.

5 Click on the **Label** tab and enter the following values:
Width and **Height**: 120,
Corner roundness: 100,
Number of labels:
Per row: 1,
Per column: 2.

6 Click on the **Page** tab and select the following values:
Page size: Letter,
Portrait.

7 Click on the **Margins** tab and enter the following values:
Margins:
Left: 40,
Top: 10,
Gap:
Between rows: 15.

8 Click on the **Stock** tab and enter the following values:
Type: Jacket,
Name: CDx2.

9 Click on OK.

Removing a customized stock

When a stock you have created is no longer needed, you can remove it.

▶ **To remove a customized stock:**

1 Choose **File Page setup...**
OR

Click on the button.

2 In the **Stock** tab, select the type to be removed if you wish to remove the type and all the stocks it contains.
OR
Select the individual stock to be removed.

3 Click on **Remove**.

Creating a Document Using Fixed Objects

3

Creating and manipulating fixed objects

Creating fixed objects involves inserting objects such as text, barcodes, images, lines, rectangles and circles into your document.

- Use the creation tools to add objects to your document, and the **Object** menu commands to define their default properties.

Note

Each object created is automatically given a name by the application. When the object is selected, its name appears in the **Document Browser Objects** tab, and in the **Object** toolbar.

Creating a Text object

A **Text** object can be one character, one word or an entire paragraph.

Character style and paragraph format can be defined using the **Text formatting** toolbar or the **Text** dialog box.

- To access the **Text** dialog box, choose **Object Text...**
OR
Press the F7 key.

▶ **To create a Text object:**

1 Click on the **Text generation** tool.

2 Click inside the workspace or on the text to be modified.
OR
Click and draw a frame within the workspace.

A flashing cursor indicates the location of the next character.

3 Enter your text.

Creating a CD sleeve: inserting text

1 Click on the **Text generation** tool.

2 Click inside the workspace and type ”Volume:”.

3 Select the text then press the F7 key.

4 Enter the following values:
Font: Arial,
Height: 10 points.

5 Click on OK.

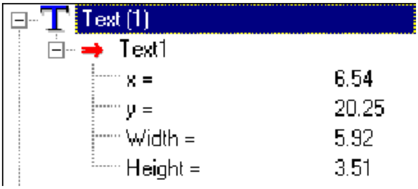
6 Repeat the above procedure to insert the text ”Conducted by:”.

▶ **To position the ”Volume:” and ”Conducted by:” objects:**

1 Select one of the **Text** objects in the document.

2 Click on the **Objects** tab in the **Document Browser**.

3 Using the mouse, open the branch of the tree marked with a red arrow (pointing to the object selected in the document).



Text (1)
Text1
x = 6.54
y = 20.25
Width = 5.92
Height = 3.51

Figure 10 Text properties accessible in the tree

For the **Text** object...	Select...	Press the F2 key, then enter...
”Volume:”	the **x coordinate:** the **y coordinate:**	”10” ”40”
”Conducted by:”	the **x coordinate:** the **y coordinate:**	”10” ”50”

4 Select the two **Text** objects and click on the **Lock** button.

Creating a Barcode object

Your software offers a wide range of symbologies adapted to the requirements of the industry, as well as the option of creating 2D barcodes.

Selecting the symbology, defining its attributes and entering the message to be encoded is carried out using the **Barcode** dialog box.

- To access the **Barcode** dialog box, choose **Object Barcode...**
  OR
  Press the F8 key.

▶ **To create a Barcode object:**

1 Click on the **Barcode generation tool**.

2 Click inside the workspace.

The **Barcode** dialog box appears.

3 Select the required barcode in the **Code** tab. You have the choice between **Printer** and **Graphic**.

4 Click on **Printer** to display the list of barcodes resident on the selected printer.
OR
Click on **Graphic** to display the list of barcodes generated by the software.

Note

Some printers do not possess resident barcodes. The **Graphic** option is used to print barcodes on these printers. However, since such barcodes are sent to the printer as graphic files, this tends to slow printing down.

5 If necessary, specify the characteristics of the barcode (height, narrow bar width, ratio, check character, and so on).

6 Enter the data to be encoded in the **Data** box, then click on OK.

Creating a Shape object

Your software comes complete with a range of tools to help you design perfect labels.

▶ **To draw a line:**

1 Click on the **Line drawing** tool.

2 Click inside the workspace.

3 Holding down the left button, move the mouse horizontally or vertically.

▶ **To draw an oblique line:**

1 Click on the **Oblique line drawing** tool.

2 Click inside the workspace.

3 Holding the left button down, move the mouse diagonally.

To...	hold down...
align the line with the 15° angle increments from the starting point,	the SHIFT key while you move the mouse.
draw an oblique line centered on the starting point,	the CTRL key while you move the mouse.
draw an oblique line centered on the starting point and aligned with the 15° angle increments,	the SHIFT and CTRL keys while you move the mouse.

▶ **To draw a rectangle:**

1 Click on the **Rectangle drawing** or **Rounded rectangle drawing** tool.

2 Click inside the workspace.

3 Holding the left button down, move the mouse diagonally.

To...	hold down...
draw a square,	the SHIFT key while you move the mouse.
draw a rectangle centered on the starting point,	the CTRL key while you move the mouse.

▶ **To draw a circle or ellipse:**

1 Click on the **Circle or ellipse drawing** tool.

2 Click inside the workspace.

3 Holding the left button down, move the mouse horizontally until the required shape is achieved.

To...	hold down...
draw a circle,	the SHIFT key while you move the mouse.
draw an ellipse centered on the starting point,	the CTRL key while you move the mouse.

Creating a CD sleeve: drawing a circle

1 Click on the **Circle or ellipse drawing** tool.

2 Click inside the workspace.

3 While holding down the left mouse button and pressing the SHIFT key, move the mouse horizontally to create the required shape.

4 Select the circle and choose **Properties...** in the context menu.

5 Click on **Position/Size...** and enter the following values: **Width** and **Height**: 20. Click on OK.

6 Choose **Object Alignment Horizontally centered** then **Object Alignment Vertically centered** to align the circle with the center of the sleeve.

7 Select the circle then click on the **Lock** button.

▶ **To draw a polygon:**

1 Click on the **Polygon drawing tool**.

2 Click inside the workspace.

3 Construct the required shape by clicking and moving the mouse.

4 Double-click to finish the shape.

To close the polygon, double-click on the starting point. You can now add a background color.

Importing an image

You software allows you to insert images and company logos to help you perfect your label.

A number of formats can be imported: bmp, dib, rle, dxf, eps, fmf, img, jpg, pcd, pcx, dcs, png, tga, tif, wmf and wpg.

▶ To import an image:

1 Click on the **Image import** tool.

2 Click inside the workspace.

The **Open** dialog box appears.

3 Select the image file to be imported, then click on OK.

To help you find the image to be imported, display the file preview by clicking on ☰.

Creating a CD sleeve: inserting the logo

1 Click on the **Image import** tool.

2 Click inside the workspace.

3 Using the **Open** dialog box, select the LOGO.BMP file located in the software installation folder: \TUTORIAL\IMAGES. Click on OK.

4 Select the logo then enter the following values:
x: 90 and **y:** 60.

5 Lock the logo.

Your label should now look like this:

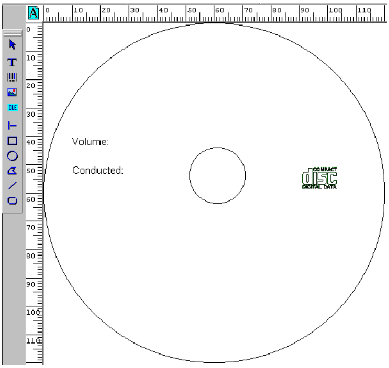


Figure 11 The example label

6 Save your document.

To finish this example, refer to Annex2 **— Connecting to Databases**.

Creating a Document Using Variable Objects

4

Your software allows you to create variable objects, the values of which can be fed by different data sources. The value of these objects, whether **Text**, **Barcodes** or **Images**, advances during printing.

Variable

Definition

A variable is a non-physical object, the value of which advances during printing. A variable can be fed in a number of different ways:

- by data extracted from an external database,
- by an external application,
- by data entered via the keyboard,
- by data calculated by the computer or printer.

Your software distinguishes between the different categories of data according to their source. You can thus create variables using a number of different **data sources**.

Data source

Database

A variable created using the **Database** data source is fed by data contained in one field in the database merged with the current document.

▶ **To create Database variables:**

Before a **Database** variable can be created, the current document must first be connected to the database. According to your requirements, you can connect to a database using either ODBC, or by directly importing an ASCII table.

First: install the ODBC data source drivers on your computer. Then install the data source, specifying the database engine and the name and access path of the database to be connected.

Connecting to a database using ODBC

1 Choose **Merge Create/Edit query...**.

2 In the **Select a data source** list, select the ODBC data source already installed.

3 Select the table(s), then the fields you wish to work with. Click on OK.

The variables are created automatically.

You are recommended to select only those fields that will actually be used in your document. This reduces the amount of data exchange with the base, thus increasing processing speed.

Importing an ASCII table

1 Choose **Merge Open ASCII table...**

2 Enter the name of the data file (*.txt).

3 Enter the name of the descriptor file (*.dsc).

4 Click on OK to import the ASCII table.

The variables are created automatically.

The Online Help contains all information relating to data file structure and defining a descriptor file.

Table lookup

The **Table lookup** data source contains a list of variables created as and when you need them. The variable values are the result of searches carried out in databases other than those linked to the current document.

Searches are carried out in one or more fields of a table in the database. The data is extracted from one of the record fields found (called the ”result field”).

▶ **To create a Table lookup variable:**

1 Select **Table lookup** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

The variable created appears automatically within the selected data source.

Note

A new sub-branch specifying the name of the data source is automatically created within the **Table lookup** branch. A new sub-branch is created automatically each time a **Table lookup** variable is created with a result field derived from a different database.

▶ **To define the properties of a Table lookup variable:**

1 Select the **Table lookup** variable, then choose **Variable properties...** in the context menu.

2 In the **Select a data source** list, select the ODBC data source in which you wish to search the field, the resulting data of which will feed your **Table lookup** variable.

3 In the **Select table** list, select the table in which the search is to be carried out.

4 In the **Select result field** list, select the field, the value of which will be transferred to your variable.

5 Define the search conditions by selecting the table field(s) in which the **Key field** column searches will be carried out.

6 In the **Key value** column, select the current document variable(s) containing the search values.

Note

The variable containing the search value must be created before the variable that will use it. When several table records meet the condition, it is the first one encountered that will be displayed.

7 Click on OK.

The **Table lookup** variable is given the same name as the table selected.

The annex entitled **Connecting to Databases** at the end of this manual provides an example of creating a CD sleeve. The example describes how to install an ODBC database, import data from a database using ODBC, create a **Table lookup** variable, carry out complex queries, and so on.

Date

The **Date** data source contains a list of variables created as and when you need them. These variables are fed by the computer or the printer's system date. They allow the date and time to be displayed, according to a predefined format. The value is updated automatically at the beginning of each print series.

You can insert the current date and/or time into a document, you can also add a fixed value to it.

▶ **To create a Date variable:**

1 Select **Date** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

The variable created appears automatically within the selected data source.

▶ **To define the properties of a Date variable:**

1 Select the **Date** variable, then choose **Variable properties...** in the context menu.

2 In the **Day** tab, click on the different elements that you wish to appear (day of the week, month, and so on).

3 To create a separator, enter it in the box using the keyboard, then click on **Add**.To use a space as the separator, click on the **Space** button.

Note Separators can be made up of several characters.

The date format is displayed in the **Date** box as you create it.

4 Click on OK.

If you wish to display the time in your document, repeat the above process in the **Time** tab. You can also combine the date and time in the same variable.

Appendix 1 at the end of the manual, Chapter 7 - 64, provides an example of creating **Date** variables.

Counter

The **Counter** data source contains a list of variables created as and when you need them. These variables are fed by data calculated either by the computer or your printer.

The content of the variable changes as printing advances, according to an initial value and an increment value.

▶ **To create a Counter variable:**

1 Select **Counter** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

The variable created appears automatically within the selected data source.

▶ **To define the properties of a Counter variable:**

1 Select the **Counter** variable, then choose **Variable properties...** in the context menu.

2 Select the type of counter.

3 Enter a value in the **Increment** box. The default increment is +1.

4 Enter a start value in the **Start value** box. The default value is 0.

5 Click on OK.

At the end of the manual there is an appendix entitled **Counters** which describes a wide range of example labels containing counters and demonstrates how to fine tune counter properties, set up complex, customized and linked counters, and so on.

Formula

The **Formula** data source contains a list of variables created as and when you need them. These variables are fed by combinations of operators, constants, variables, control variables, formulas and functions. Data can be both numerical and alphanumerical.

▶ **To create a Formula variable:**

1 Select **Formula** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

The variable created appears automatically within the selected data source.

▶ **To define the properties of a Formula variable:**

1 Select the **Formula** variable, then choose **Variable properties... in** the context menu.

2 Enter the formula directly in the edit box.
OR
Select the elements of your choice with the mouse, then click on **Insert**.

More advanced! You can insert an element simply by double-clicking on it.

3 Click on **Test** to verify that the syntax is correct. If an error occurs, follow the instructions displayed on the screen and carry out any necessary changes.

4 Click on OK.

If the name of a variable used in the formula contains any of the following characters: &+-*/<>=^%,!\”, the name must be enclosed in {}s.

At the end of this manual is an appendix entitled **Formulas** offering a wide range of example labels containing formulas. The examples demonstrate how to fine-tune formula properties, set up complex formulas, calculate specific ”modulos”, and so on.

Form

The **Form** data source contains a list of variables created as and when you need them. These variables are fed by entering data directly into the **Form** via the keyboard.

▶ **To create a Form variable:**

1 Select **Form** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

The variable created appears automatically within the selected data source.

▶ **To define the properties of a Form variable:**

1 Select the **Form** variable, then choose **Variable properties...** in the context menu.

2 Ensure the **Display in form** box is checked.

3 Enter the prompt text you wish to display in the **Form** using the **Prompt** box.

4 Enter an input syntax (for example: ##-##-## if it is a date that will be entered).

5 In the **Minimum length** box, enter the minimum number of characters that must be entered for text to be validated. You can also enter the values that will be suggested in the **Form** by checking the **Pick list** box.

6 Click on OK.

The Form

The **Form** allows you to enter data directly.

It is possible to change the appearance of the form and make it more readable by changing the color, adding borders or changing the font, and so on. The customized form will be saved with your document.

- To access the **Form**, choose **View Mode Form**,
  OR
  Click on the **Open form** button on the **Display** toolbar.
- To change the appearance of the **Form**, click inside the **Form** and choose **Form customization...** in the context menu,
  OR
  Choose **Format cells...** to change the color, style, form borders, and so on.

The **Form customization** dialog box offers the following options:

Option	Definition
Display rows	displays the rows.
Display row headers	displays the row numbers. In conjunction with **Enable row moving**, this option allows you to change the order that data appears in the **Form**. Select the number of the row to be moved, then drag the mouse to the new location.
Lock form	prevents modification, moving, resizing or deleting of rows.
Distribute rows evenly	adjusts row height to the minimum necessary to display the tallest row.
Enable row moving	allows rows to be moved within the form, providing the **Display row headers** option is activated.
Display form when printing	offers three options that come into effect when printing is launched: - **No**: the **Form** will not be displayed. - **For each label**: the **Form** will be displayed for each label in a print run. - **For each series**: the **Form** will be displayed once at the beginning of a print run.

Free

The **Free** data source contains a list of variables created as and when you need them. A variable is called **Free** if the origin of the data that feeds it is not specified.

For example, a **Free** variable can be useful for creating intermediate variables used for creating **Formula** or **Table lookup** variables.

▶ **To create a Free variable:**

1 Select **Free** in the **Document Browser Data Sources** tab.

2 Choose **Add...**in the context menu.

The variable created appears automatically within the selected data source.

▶ **To define the properties of a Free variable:**

1 Select the **Free** variable, then choose **Variable properties...** in the context menu.

2 Select one of the following options: **Local** or **Shared**.

The Online Help contains further information on shared variables.

3 If you have chosen the **Local** option, enter the variable value in the box. This is the initial value of the variable, the one that will be printed first.
OR

If you have chosen the **Shared** option, click on the [...] button if you need to modify the current value of the variable. Click on OK.

4 Click on OK.

Specific examples of using **Free** variables can be found in the **Formula** and **Connecting to Databases** appendices.

Variable objects

The previous chapter introduced the concept of objects. We are now going to take this one stage further and look at variable objects.

A variable object is a variable that takes on a physical appearance, whether text, barcode or image. Thus, the value of any variable from any source can be displayed in your document in the form of text, or encoded in a barcode.

Moreover, several objects can be associated with a single variable.

Creating variable objects

Creating a variable object involves associating an object with a document variable.

There are two ways of creating variable objects:

- creating a variable object from a fixed object;
- creating a variable object using a creation tool.

▶ **To create a variable object from a fixed object:**

1. Select a **Text** object in the document (the whole object must be highlighted).
2. Select **Properties...** in the context menu.
3. Click on the **Data source** tab.
4. Select the **Variable data** option, then select the data source that will feed the variable object.
5. Click on OK.

Do not forget to click on the **Display field contents** button to view the variable values.

▶ To create a variable object using a creation tool:

1. Click on the **Text generation**, **Barcode** or **Image import** tool.
2. Select the variable you wish to associate with the object in the **Document Browser Data Sources** tab.
3. Drag and drop the variable to the required place in the workspace.

   The variable object thus created displays the name of the associated variable.
4. To display the variable value, click on the **Display field contents** button.

More advanced!

1. Select the variable you wish to associate with the text object in the **Document Browser Data Sources** tab.
2. Drag and drop the variable to the required place in the workspace.

   A context menu appears, listing the different objects.
3. Select the required object.

Note

The variable must be created before the variable object.

Interchanging variables:
To replace a variable already inserted into your document with another variable:
- Select the variable to be replaced, then the select the substitute variable in the **Document Browser**.
- Drag and drop the new variable onto the one to be replaced. The new variable appears instantly.

Printing

5

A powerful print engine

Your software is both a powerful, user-friendly creation tool and a **print engine**, bringing together the widest possible range of printers (thermal, thermal transfer and inkjet), never before available on the automatic identification market.

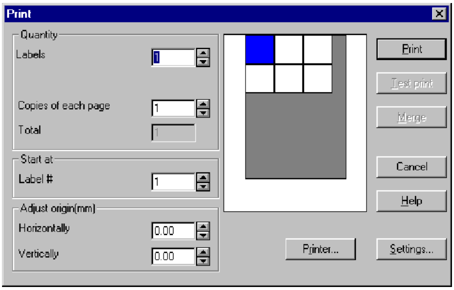
A number of solutions are available for printing your documents: individually, in batches, and so on.

Traditional printing

Whether printing a single label, a series of labels, or multiple copies of the same label, all print settings are made using the same dialog box: **Print**.

1 Choose **File Print...** or click on 🖨 or press the **F6** key.

The **Print** dialog box appears on the screen.



Print

Quantity

Labels: 1

Copies of each page: 1

Total: 1

Start at

Label #: 1

Adjust origin(mm)

Horizontally: 0.00

Vertically: 0.00

Figure 12 The **Print** dialog box

2 Click on **Print**.

The current label (the one displayed on the screen) is printed.

If the document includes a counter, the number entered in the **Labels** field allows you to determine the number of **different** labels to be printed, i.e. the number of counter increments.

If your document does not contain a counter, you can print multiple copies of the same label by entering the required number in the **Labels** box.

Practical Workshop 1 - Label Series and Label Copies

First of all, we need to distinguish between **Labels in the series** and **Copies of each label**.

Note

Note that if the label does not include a counter, the **Print** dialog box will not display the **Copies of each label** field, but only the **Labels** field, which in this case serves the same purpose.

To highlight the difference between **Labels in the series** and **Copies of each label**, we shall consider a label containing two types of counters, and configure the **Print** dialog box in different ways.

- Open the TUTORIAL\LABELS\COUNTER.LAB label.

This is a simple label containing two different counters:

- a classical counter that counts all labels, and resets when its value reaches 10:
  - Decimal
  - Increment +1
  - Start value 0
- an ISO counter allowing each label copy to be individually and exhaustively numbered. It also resets when its value reaches 10:
  - Decimal
  - Increment +1
  - Start value 0

The first counter, called **NumLabel**, identifies a particular label, whereas the ISO counter counts the total number of labels printed.

Labels in the series

In this example, we are going to print a series of 10 labels:

1 Enter the following values in the **Print** dialog box:
Labels in the series: 10.
This means that there are 10 different labels in the series.

2 Click on **Print**.

The printed output is as expected: 10 labels have been printed and the two counters read the same value.

Copies of each label

Here we will print a series of five labels, but this time we want two copies of each one.

1 Enter the following values in the **Print** dialog box:
Labels in the series: 5
Copies of each label: 2.

2 Click on **Print**.

10 labels have been printed, however the result is totally different from the one obtained in the last example (see **Figure 13** below).

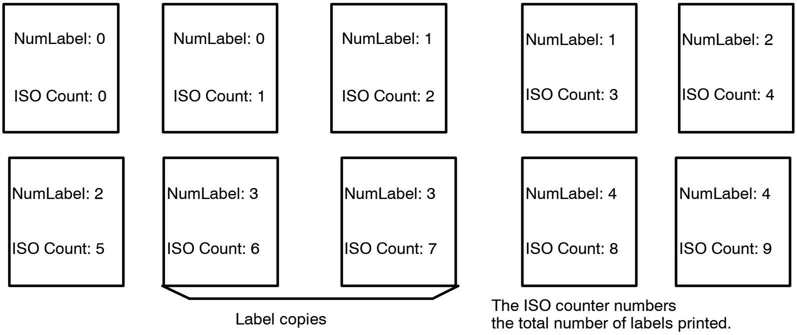


Figure 13 The printed output

Copies of each page

As we have just seen, it is possible to print multiple copies of the same label. In the next example, we will see that it is also possible to print multiple copies of the same page. We therefore need to know exactly what the printer understands as being a page.

The page is determined by the parameters specified when the printer driver is configured. The results obtained may be completely different depending on whether you print on an office printer or a thermal printer. It can be quite difficult to visualize a page on continuous paper, as used mainly by thermal printers.

Moreover, according to the dimension of the print media and the size of the label, the page dimensions may correspond to those of the label, and thus, by extension, the page copy may be the same as the label copy.

The page format and label size are defined in the **Page setup** dialog box.

1 Open the TUTORIAL\LABELS\COUNT01.LAB label.

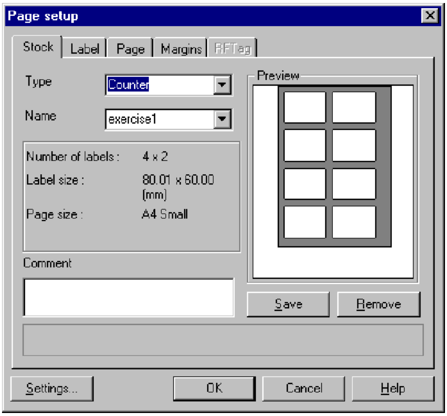
2 Choose **File Select printer...**

3 Select a Windows office printer.

4 Make sure that the paper format is A4, then click on OK.

5 Choose **File Page setup...** or click on 🖨 .

The **Page setup** dialog box appears.



Type: Counter

Name: exercise1

Number of labels : 4 x 2

Label size : 80.01 x 60.00 (mm)

Page size : A4 Small

Figure 14 The **Page setup** dialog box

The preview gives you an idea of how the page will look when it is printed.

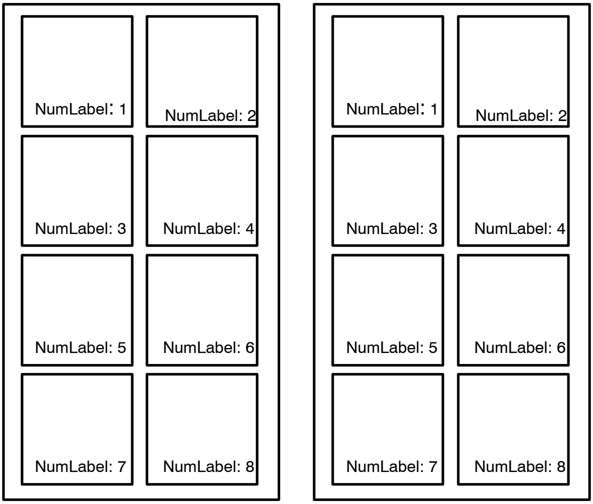
6 Close the dialog box.

7 Click on 🖨 .

8 Enter the following values:
Labels in the series: 8
Copies of each page: 2

9 Click on **Print**.

The result is as follows:



Warning! The same label printed on a thermal printer using continuous paper may give a completely different result.

Figure 15 An example of printed output

If you label contains an ISO counter, **Page copy** will be the same as **Label copy**.

Practical Workshop 2 - Print Merge

Print **Merge** is only available if your label is linked to a database (ASCII or ODBC).

The **Merge** function prints all the records contained in the database.

Print Merge

1 Open the TUTORIAL\LABELS\IDENTITY.LAB label.

2 Make sure that the connection with the TUTORIAL\LABELS\IDENTITY.MDB database is correctly set up (see **Chapter 8 - 73**).

3 Choose **File Print...** or click on or press the **F6** key.

The **Print** dialog box appears.

4 Click on **Merge**.

All records in the database are printed, i.e. three labels.

Practical Workshop 3 - Printing Using the Form

It is often the case that variable data needs to be entered by an operator just prior to printing. Your application offers a quick and simple method of data entry: the **Form**.

Customizing the Form

You are free to customize forms by choosing the background and border color for each cell in the grid, font type, font size, and so on.

1 Open a new document.

2 Add a variable using the **Form** data source in the **Document Browser**.

3 Rename it **NAME** (F2 key), then double-click on it.

4 Complete the dialog box with the following information:
- ensure the **Display in form** box is checked;
- enter ”Name” in the **Prompt** field.

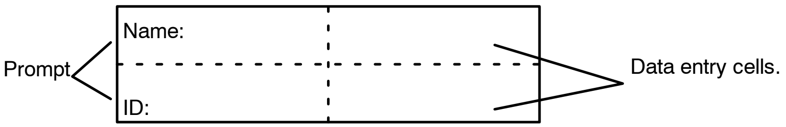
5 Click on OK.

6 Repeat the above procedure to create a variable called **ID**. Complete the dialog box with the following information:
- ensure the **Display in form** box is checked;
- enter ”ID” in the **Prompt** field.

7 Insert them into your document.

8 Display the **Form** either by choosing **View Mode Form**, or by clicking on or by pressing **F4**.

The **Form** appears, as shown in **Figure 16** below:



Borders are not displayed by default. The dotted lines mark the different cells in the **Form**.

Figure 16 The default **Form**

9 Select a cell and display the context menu (right mouse button).

The following two commands are displayed: **Format cells...** and **Form customization...**

Format cells allows you to define:

- The font type and color for displaying the prompt.
- The cell background color.
- Cell border color.
- Text alignment with the cell.
- Set the above options as required.

Form customization allows you to:

- Display row headers (once row headers are displayed you can select an entire row and change its position).
- Display the form as a grid.
- Configure the way the **Form** behaves, i.e. whether it is displayed **For each label** or **For each series**. For the purposes of our example, select the **For each label** option for both variables.

Printing using the Form

Now that the **Form** looks the way you want, we will launch printing.

1 Choose **File Print...** or click on 🖨 or press the **F6** key.

2 Enter ”2” in the **Labels in the series** box.

3 Launch printing.

The form is displayed, and you are invited to enter the data to be printed.

4 Enter ”Smith” and ”51”, for example, then click on OK.

Once the first label is printed, the form is displayed again.

5 Enter ”Mullang” and ”52”, for example, then click on OK.

Note

The **Form** is displayed for each label contained in the series. If you choose the **For each series** option, the **Form** will only be displayed once, when printing of the label series is launched.

Summary

The Print dialog box contains differences...	When the label contains a counter: **Labels in the series** allows you to specify the number of different labels in a series; **Copies of each label** allows you to determine the number of copies of the same label to be printed. When the label does not include a counter, this option serves the same purpose as the **Labels field.**
Page copies	A page copy can only be carried out if the label does not include an ISO counter and if the page contains more than one label.
Print Merge	This option prints all the records in a database, and is only available when the label is connected to a database.
Printing Using the Form	The **Form** allows an operator to enter data by keyboard just prior to printing.

Optimizing Printing

An increasing number of printers come equipped with a parallel port, and we would recommend using this rather than the serial port. If you envisage printing bitmap images, graphics or Windows™ TrueType fonts, using the parallel port will greatly enhance your system's performance in terms of data transfer capacity. Moreover, there is only one type of cable for all parallel links, whereas there is no standard for serial cables. You thus avoid any problems over printer cable requirements.

That said, given that many printers are still supplied with a serial link only, this section also covers most of the problems you are likely to encounter with serial communications, and explains how to configure Windows™ for trouble-free printer communications.

Furthermore, it is important to bear in mind that Windows™ printer drivers, and the program you use for printing, do not control serial communications between your computer and the printer. The communication mode is controlled by:

- Your printer's settings.
- The type of serial cable used.
- The serial communication settings used in the Windows™ control panel.

Optimizing printing speed

As we have already seen in the introduction, there is no universal configuration for optimizing printing speed, rather a range of different settings that come into play. In the next example we will look at each of these in turn, hopefully giving you a better understanding of your own configuration, and thus enabling you to find the best settings for optimized printing.

For improving printing speed, there are four basic principles we need to bear in mind:

- use the parallel port wherever possible;
- use printer objects (fonts, barcodes, and so on) wherever possible;
- in the case of a serial link, and if the printer offers the option, make use of the **Windows Font Downloader**. This plug-in tool allows you to download any Windows™ font into the printer's memory—quickly and simply—thus turning it into a printer resource. For more information on this plug-in (provided with your application), refer to the documentation available on the application CD-ROM;
- use the ”**Send modified data only**” command, available for the vast majority of printers and available in the Printer settings dialog box;

The **Direct access** option is offered when configuring the printer driver and selecting the port. The table below lists when this option may be used:

Port	Direct access	Non-direct Access
Serial	yes	yes
Parallel	yes	yes
Network	no	yes

Using this option will depend on your particular requirements :

- With **Direct access**, the printer driver communicates directly with the printer: the user has no control over events until printing is completed (possibly involving something of a delay);
- Without the **Direct access** option, the printer driver transfers data to a print queue, which manages communication: the user regains control once all data has been transferred to the print manager (involving much less of a delay).

▶ Parallel link and serial communication

For printing images, use the parallel link wherever possible.

Given the many advantages of using images, such as the wide variety of Windows graphics, you will get significantly better results by using a printer equipped with a parallel communications port. The majority of standard thermal printers are equipped with a serial link only (RS 232), with the parallel port as an optional extra.

Serial communication works fine, as long as you are only intending to print printer objects (barcodes, fonts, and so on). If you wish to print bitmap images, printing speed will be greatly reduced.

▶ For printing images...

Improved performance for printing a number of label series

The label background is stored in the printer's memory. As a result, the fixed image does not need to be transferred via the cable once the first label is printed.

For example, your company logo, not a printer object, must be transferred to the printer. The more images you send to the printer, the slower printing becomes—especially if you are using a serial link. The solution is to send the image in question just once, when printing of the label series is first launched.

The majority of printers allow this operation.

Use the **Send modified data only** option: the image is sent to the printer only once, when printing starts, which can take a little time. However, once the first label is printed the data transfer process runs much faster.

Troubleshooting

The table below lists the problems most frequently encountered during printing, and offers some solutions to help you resolve them.

Problem	Cause	Solution
Nothing is printed	The printer is not plugged in.	Plug the printer in.
	For a parallel link : the printer is not configured for a parallel link.	Make sure that this setting has been configured in the printer control panel (refer to the manufacturer's handbook).
	For a serial link: - the serial communication settings are incorrect.	Make sure that the communication settings between the PC and the printer are the same (for the serial communication settings required by the printer, refer to the manufacturer's handbook).
	- the serial cable is not suitable.	Make sure that the cable matches the specifications of your printer (refer to the manufacturer's handbook).
	In both cases (serial or parallel): - the cable is connected to the wrong port.	Check the connection.
Page is not printed	For direct thermal printing : the print media has been inserted the wrong way round (the print head is heating the wrong side of the paper).	Reinsert the media the right way round.

Problem	Cause	Solution
Page is not printed (continued)	For thermal transfer printing : the ribbon has not been fitted, or has been fitted incorrectly.	Check the ribbon and refit if necessary (refer to the manufacturer's handbook).
	The required print mode has not been selected on the printer.	Check that the print mode (thermal or thermal transfer) has been correctly selected on the printer (refer to the manufacturer's handbook) or in the Printer setup dialog box.
A graphic object (image, Windows font, graphic barcode, etc.) has not been printed.	The printer is out of memory.	Check the **Clear memory at next print time only** option; if the problem persists, check amount of memory allocated to images (**Images** tab of you printer driver).
	The printer does not have enough memory.	Increase the size of the printer's memory either by adding an extra memory card, or by increasing the RAM (refer to the manufacturer's handbook or the dealer where you purchased the printer).
	Certain printers will not recognize an object if it overlaps the edge of the printable area, even by only a few millimeters.	Readjust the position of the object within the label.
The thermal printer is jammed	The print media used is not the same as the one defined in the **Page setup** dialog box (continuous paper or with gap).	Check the right print media was selected during page setup: Continuous, With gap.

Security Made to Measure

User Manager is a tool that allows you to manage access rights to the label design software. It also allows you to set up group work strategies by organizing user accounts into user groups. You can create and/or remove user accounts with **User Manager**.

Note

Any changes made to user access strategies will take effect the next time the user runs the application.

There are many advantages to a strategy of working in user groups. Managing user accounts in this way allows you to adapt the application to the requirements of your company, adding extra levels of security according to the different services, thus guaranteeing the integrity of your documents and data.

You can use separate accounts for creating, modifying or simply printing documents.

User Manager

User Manager allows you to manage access rights to the various documents by means of user accounts. The administrator can create as many accounts as required, each one having its own specific rights. Users can be added to one or more accounts.

An account entitled **Guests** is created automatically when **User Manager** is launched. It contains no default access rights.

Note

Adding the security plug-in disables all commands linked to the **Password** function.

Note

User Manager can only be activated once it has been configured by the administrator.

▶ A user-friendly interface for optimized management of access rights...

The interface has been designed to make administering the different user groups as simple as possible. Presented in the form of a tree, like Windows™ Explorer, and using context menus, drag and drop, and so on, working with User Manager is totally intuitive. In **Chapter 6 - 51**, we will take a closer look at how the **User Manager** interface works.

The administrator

User Manager cannot be run without an administrator. The administrator alone is authorized to manage the various groups: assigning and removing access rights, adding users or other administrators, creating groups, and so on.

▶ Who is the administrator?

The administrator is the one who runs **User Manager** for the first time.

Note

An administrator is the only one who is authorized to add or remove another administrator.

Without profile management

The computer on which your label design software and **User Manager** are installed does not manage user profiles.

In this case, each time the label design software is launched, a dialog box is displayed asking for a user name and password.

If **User Manager** does not recognize the user, i.e. he/she is not registered in one of the user groups, he/she will automatically be defined as a **Guest** and will thus only be granted those rights the administrator has assigned to the **Guests** account.

With profile management

The computer on which your label design software and **User Manager** are installed does manage user profiles.

In this case, **User Manager** administers the user profiles declared on the network. When the label design software is launched, **User Manager** checks whether it recognizes the user, i.e. that the user is registered in one of the user groups. If not, the user will automatically be defined as a **Guest** and will thus only be granted those rights the administrator has assigned to the **Guests group** as assigned by the administrator.

Note

If the workstation on which your label design software is installed is connected to an NT network, the administrator will have direct access to the names of all users on the network. This reduces the risk of creating users with incorrect names.

Administration

In this section, we will look at all the various functions available in **User Manager**.

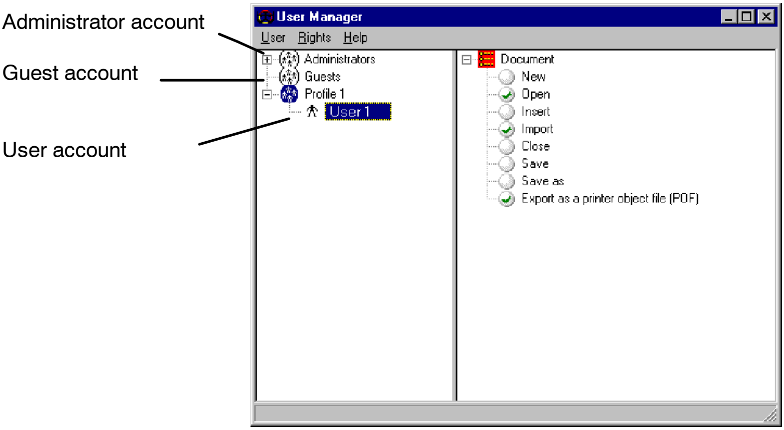


Figure 17 User Manager

Adding a user

▶ **To add a new user to an existing profile:**

- Choose **User New User**.

  A new branch is created within the profile, allowing you to enter the name of the new user, see below:

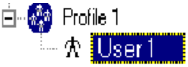


Figure 18 Adding a new user

Adding a new profile

Adding a new profile allows you to define the various rights associated with a particular area of activity within your organization.

▶ **To add a new profile:**

1 Choose **User New profile**.

The new profile is thus created.

2 Give it a name, and configure its rights, see below.

Note By default, a new profile is assigned all rights.

▶ **To define a new profile:**

As we have already seen, when a new profile is created, all rights are assigned to it. The administrator can then remove those rights that do not apply to the profile in question by simply clicking on them with the mouse.

▶ **To delete a group of rights:**

- Select the group of rights to be deleted then choose **Rights Delete**.

  The group disappears from the tree.

▶ **To delete specific rights:**

- Click on the group of rights to be modified then click on the individual rights one by one with the mouse to disable them.

▶ **To move a user from one account to another**

- Select the user, then drag and drop it into the new account.

Appendix 1– Formulas

7

Calculations with Formulas and Functions

Your software offers a wide range of calculation functions, allowing you to carry out date calculations, character string manipulations, and so on. A formula can contain variables, constants, functions, operators, and so forth.

To carry out such calculations within a document, you will need to create a variable: the **Formula** variable.

With its own specific dialog box, this variable allows you to define the required function(s) for a given formula.

You can create as many **Formula** variables as there are calculations to be carried out in your document.

This appendix is complemented by numerous examples that will help you to put the various types of formula into practice. The examples entitled **Practical Workshop** come in different levels of difficulty, as indicated by the following icons:

😄	**Very easy**	🙂	**Easy**
😐	**Difficult**		

The Different Types of Formulas and Operators

Numerical formulas

Calculations are carried out on numerical values and give numerical results. Values can be either variables or constants.

2*Var0 = 14
The variable **Var0** refers to the value of a batch number; **2** is a constant. The result varies according to **Var0**, but is always the value of **Var0** multiplied by **2**.

Logical formulas

Logical formulas allow you to check whether or not one or more conditions have been met.

Var0>Var1
This formula checks whether the value of **Var0** is greater than that of **Var1**. If yes, the value returned will be 1 (true), if not it will be 0 (false).

Text formulas

Text formulas involve characters that are, strictly speaking, not part of the calculation, but which generate the character strings.

”NAME: ”&Var0
The result obtained is **NAME: Smith** if **Smith** is the value of **Var0**.

About functions

Functions are ready-to-use formulas, which sometimes look rather complicated. A function can be integrated into a formula.

A function uses values and arguments to generate new values. It is not necessarily a case of performing a calculation in the strict sense of the word, rather logical analysis or manipulation of character strings, for example.

There are six groups of functions in the formula definition :

- Check digit calculation functions.
- Conversion functions.
- Date and time functions.
- Logical functions.
- Mathematical functions.
- Character string functions.

Operators Calculations are carried out in a strict order of priority, which is why it is sometimes necessary to enclose certain elements of the formula in brackets to indicate, or change, the order in which operations are carried out.

Operator	Example	Meaning	Priority
Arithmetical operators			
^	5^2 = 25	exponent	1
%	20%3 = 2	modulo	2
*	5*2 = 10	multiplication	3
/	6/2 = 3	division	4
+	20+5 = 25	addition	5
–	20–5 = 15	subtraction	6
Text operators			
&	”string1”&”string2”	concatenation of two strings	7
Comparative operators			
=	Var0=Var1	equals	8
>	Var0>Var1	greater than	9
<	Var0<Var1	less than	10
>=	Var0>=Var1	greater than or equal to	11
<=	Var0<=Var1	less than or equal to	12
<>	Var0<>Var1	different from	13

The priority rating indicates how the application works when different operators are combined in the same formula. In the following formula:

4+5*6-9/2
First the multiplication is carried out (5*6), followed by the division (9/2), then the addition (4+30), and finally the subtraction. The result would be remarkably different if the correct order of priorities had not been followed. The result of the calculation is 29.5, whereas if the calculation had been carried out in the order it is written, the result would be 22.5. When two operators have the same priority rating, the calculation is carried out from left to right. Brackets allow you to isolate certain calculations. For example:
(4+5)*(6-9)/2
Gives -13.5, and not 29.5.

Creating a Formula

The value of a **Formula** variable is determined by a combination of operators, constants, variables, formulas and functions. The result can be numerical, alphabetical or alphanumerical.

The Formula dialog box

The **Formula** variable dialog box allows you to enter the expression. The dialog box is divided into three tabs:

- **Formula**: allows you to determine the characteristics of the formula;
- **Output**: allows you to adjust the final formatting of the variable for display in your document (see Online Help **Output tab**);
- **Counter**: allows you to assign the variable a **Counter** attribute (see Online Help **Counter** tab);

▶ **To access the Formula variable dialog box :**

1 In the **Data Sources** tab of the **Document Browser**, select **Formula**.

2 Choose **Add...** in the context menu.

A sub-branch called **Formula0** appears.

3 Select the sub-branch, then double-click on it.

The **Formula** dialog box appears (see Figure 19).

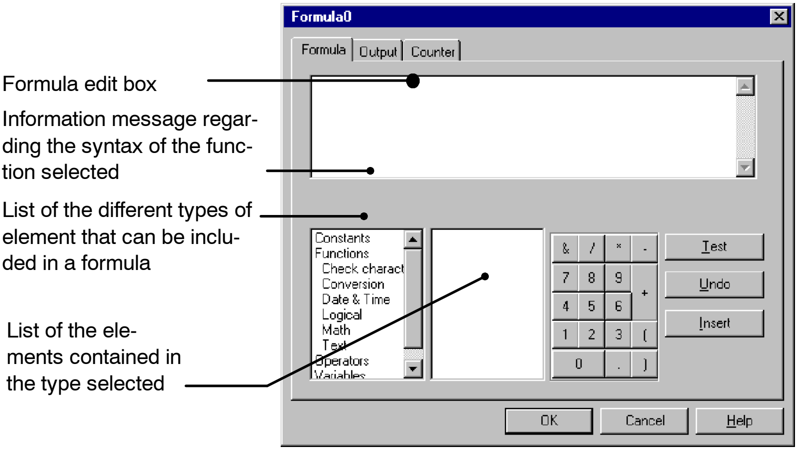

Formula edit box
Information message regarding the syntax of the function selected
List of the different types of element that can be included in a formula
List of the elements contained in the type selected
Formula0
Formula
Output
Counter
Constants
Functions
Check charact
Conversion
Date & Time
Logical
Math
Text
Operators
Variables
&
/
*
-
7
8
9
+
4
5
6
1
2
3
(
0
.
)
Test
Undo
Insert
OK
Cancel
Help

Figure 19 The **Formula** dialog box

Defining the formula

The first step involves creating the expression that defines the formula.

You can enter elements into the formula edit box either by typing them in directly using the keyboard, or by selecting them on screen with the mouse.

▶ **To enter the expression:**

1. Select the elements that make up the expression one by one, clicking on **Insert** to add them to the expression.
2. Once you have finished entering the expression, click on the **Test** button. This allows you to check whether there are any errors in the syntax of the expression.
3. Click on OK.

▶ **To rename the variable:**

- To rename a variable, select it, click on it once to select the default name, then enter the new one.
  OR
  Press the F2 button and enter the new name.

▶ **To create the variable object and insert the Formula into the document:**

- Click on the **Text generation** tool, select the formula then drag and drop it into the workspace.

Do not forget to select the **Content** view mode in order to check the result of your formula.

If the result is truncated, check the settings in the **Output** tab: the maximum length value may be too low to display the full result of your formula.

The Online Help provides an exhaustive list of all the functions, and provides explanatory notes and examples specific to each one.

Practical Workshop 1 – Complex Formulas

Logical functions

You can use logical functions to check whether or not one or more conditions have been met. For example, you can use the **if** function to determine whether or not one condition has been met: one value will be returned if the condition has been fulfilled, another if it has not.

The example below presents a document designed for labeling boxes. It contains a name and address, plus information on the contents. All boxes for shipping need to be numbered, up to and including the total number of boxes in the series, a value determined in advance. Box numbers are entered manually using the **Form** when printing is launched. To check for possible errors, a formula displays an "ERROR" message when the box number (BOX# variable) is greater than the total number of boxes in the series (TOTAL # OF BOXES variable).

1 Open the overrun.LAB label, located in the TUTORIAL folder.

2 Choose **File Print...** then launch printing by clicking on the **Print** button.

The **Form** appears. We will now enter a value for the **BOX#** variable deliberately greater than the **TOTAL # OF BOXES** variable, set at 11.

3 Enter "12", then click on OK in the **Form**.

You will notice that the printed label contains the message "ERROR" (see Figure 20).

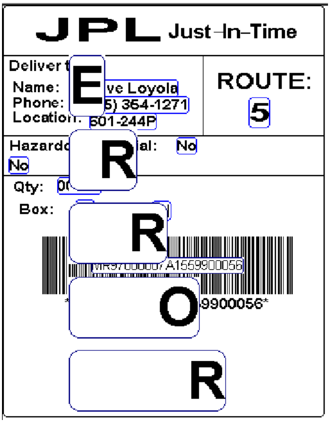


Figure 20 The Example Label

The formula that displayed this message is the following:
if(value(BOX #) >value(TOTAL # OF BOXES),"E\n R\n R\n   O\n   R","")
(/n: new line)

▶ The IF function

Returns one value if the condition you specify is TRUE, and another value if it is FALSE.

Use the **if** function to carry out a conditional test on values and formulas.

▶ Syntax

if("expr","val_if_true","val_if_false")

"expr" represents any value or expression, the result of which can be TRUE or FALSE.

val_if_true is the value returned if "expr" is TRUE. The val_if_true argument can be another formula.

val_if_false is the value returned if "expr" is FALSE. The val_if_false argument can be another formula.

Note

It is quite possible to link a number of **if** functions such as "val_if_true" and "val_if_false" to create more complicated tests (see example below).

```
if(dayofmonth > 10,
     if(dayofmonth > 15,
          if(dayofmonth >25, "Risk of being cut off!","Pay
          the bill now!"),
          "Think about paying the bill!"),
"Everything's fine!")
```

Or, in English:

```
"If dayofmonth > 10 then
        If dayofmonth > 15 then
                If dayofmonth > 25 then
                        Risk of being cut off!
                If not
                        Pay the bill now!
        If not
                Think about paying the bill!
If not
        Everything's fine!"
```

This example tests the day of the month:
- If the day of the month is between 1 and 10, the message ”Everything's fine!” is displayed;
- if the day of the month is between 11 and 15, the message ”Think about paying the bill!” is displayed;
- if the day of the month is between 16 and 25, the message ”Pay the bill now!” is displayed;
- if the day of the month is greater than 25, the message ”Risk of being cut off!” is displayed.

If you wish to test the above formula, enter it in the formula edit box. Do not forget to create the intermediate variable **”dayof-month”** first.
This intermediary variable is created in the **Free** branch (**Data Sources** tab of the **Document Browser**).
In order that messages are displayed in their entirety, enter a value of at least 22 in the **Maximum length** box in the **Formula** dialog box **Output** tab.

The Online Help provides examples for all the logical functions available in your software (see **Online Help Logical Functions**).

Practical Workshop 2 – Calculating a Specific ”Modulo” 😐

Your software comes with integrated modulo calculation functions for generating barcode check characters.

However, it may be that you need to calculate a specific check character, and, as a result, the integrated functions do not meet your requirements.

You will thus need to create the function yourself.

In the next sequence we will see how it is possible to use formulas to create any kind of check character calculations.

This example illustrates how a control character can be calculated for a **2/5 interleaved** barcode.

Method for calculating a check character

The method involves multiplying the first character of the data string by 1, the second by 2, the third by 1, and so on.

- Open the label entitled CHECK01.LAB, located in the TUTORIAL folder.

The check character calculation will be carried out on the **LOT NUMBER** DATA.

▶ To calculate the weight

1 Click on the **Formula** branch of the **Document Browser Data Sources** tab, and add a new **Formula**.

2 Enter the following formula:
mid(LOT NUMBER,1,1)&
mid(LOT NUMBER,2,1)*2 &
mid(LOT NUMBER,3,1)&
mid(LOT NUMBER,4,1)*2 &
mid(LOT NUMBER,5,1)

The result is 2120103; so we have:
2*1 = 2
6*2 = 12
0*1 = 0
5*2 = 10
3*1 = 3
The concatenation of the results gives 2120103.

3 Name the formula ”WEIGHTED”.

▶ To add the result of the weight calculation:

The following step involves adding together the figures resulting from the previous formula.

1 Create a second **Formula**, and name it ”SUM”.

2 Enter the following expression in the text box:
mid(WEIGHTED,1,1)+mid(WEIGHTED,2,1)+
mid(WEIGHTED,3,1)+mid(WEIGHTED,4,1)+
mid(WEIGHTED,5,1)+mid(WEIGHTED,6,1)+ mid(WEIGHTED,7,1)

The result is 9; so we have:
2+1+2+0+1+0+3= 9.

▶ **To calculate the check character:**

Using the previous result, we will calculate the value of the check character.

1 Create a third **Formula**, and name it ”CHECK DIGIT”.

2 Enter the following expression in the text box:
if((SUM % 10)>0,10-SUM % 10,0)

The result is 1; so we have:
sum % 10 = 9 (% = modulo).
9 being greater than 0, we carry out the subtraction:
10- 9 = 1.

Note

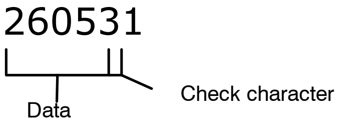
If the result had been equal to 0, the value of the check character would have been 0.

▶ To calculate the data to be encoded:

Creating the barcode must include the data to be encoded, i.e. the value of the "LOT NUMBER" VARIABLE concatenated with the check character (CHECK DIGIT).

1 Create a fourth **Formula**, and name it "DATA".

2 Enter the following expression in the text box:
LOT NUMBER & CHECK DIGIT.
The data to be encoded is now:

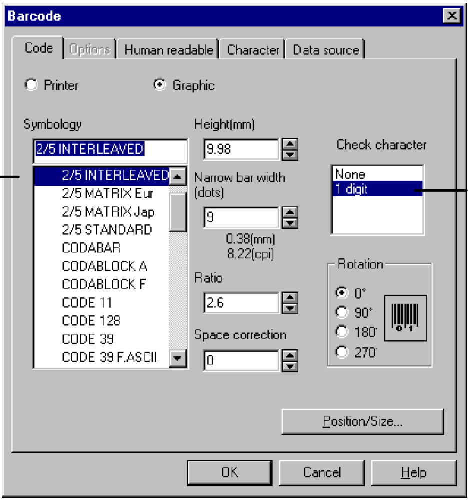


▶ To create the barcode:

1 Click on the **Barcode** tool located on the creation toolbar. Next, select the **DATA** formula then drag and drop it into the label.

2 Select the barcode and choose **Object properties...** in the context menu.

The **Barcode** dialog box appears (see **Figure 21** below).



Selecting the **2/5 interleaved symbology**

Presence of a check character

○ Printer ● Graphic

Symbology: 2/5 INTERLEAVED

Height(mm): 9.98

Narrow bar width (dots): 9
0.38(mm)
8.22(cpi)

Ratio: 2.6

Space correction: 0

Check character: 1 digit

Rotation: ● 0° ○ 90° ○ 180° ○ 270°

Figure 21 The **Barcode** dialog box

3 Configure the dialog box so that it looks like **Figure 21**.

The label is now finished. The label stock is located in TUTORIAL\CHECK.LAB.

Practical Workshop 3 – Date Calculations☺

Calculating an expiry date using the system date

Your software includes an option that allows you to add a fixed or variable time period to the system date. This function is extremely useful for calculating the use by date of a product, for example.

We will now create a label containing two **Date** variables. The first for the date of manufacture, the second for the expiry date.

▶ **To create the Date variables:**

1. Using the **Document Browser Data Sources** tab, select **Date**.
2. Click on **Add...** in the context menu.

   A sub-branch **Date0** appears.
3. Select it and press the F2 key to rename it.
4. Enter **DATE OF MANUFACTURE**.
5. Select it again, and double-click.

   The **Date** dialog box appears (see Figure 19).
6. Click on **Day of the month** ″**01 - 31**″.
7. In the edit box in the lower left of the dialog box, enter a separator, the ″/″ character, for example, then click on **Add**.
8. Next, click on **Month** ″**01 - 12**″, add the separator then click on **Year** ″**00**″.
9. Click on OK.

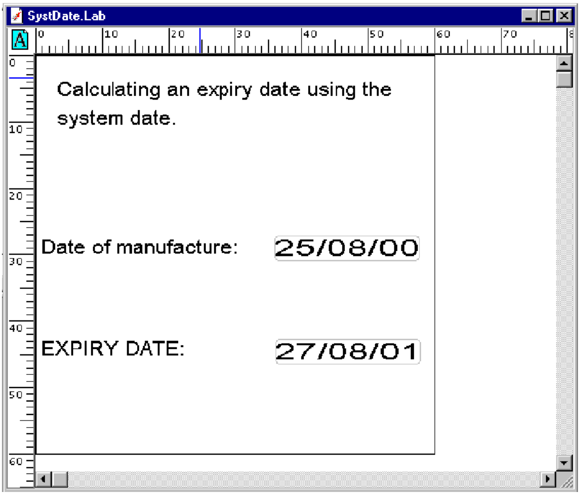
We now need to create the **EXPIRY DATE** variable:

1. Repeating the above procedure, create another **Date** variable, and name it **EXPIRY DATE**.
2. Bring up the **Date** dialog box by double-clicking on **EXPIRY DATE**.
3. As for the **DATE OF MANUFACTURE**, insert the **Day of the month**, **Month** and **Year**, plus the separators.

Let us say that the shelf life of this product is one year and two days, i.e. 367 days. After that, the product will no longer be fit for human consumption.

4. Select the **Options** tab, check the **Fixed** option in the **Add a value** box and enter **367**.
5. Select **Days** as the **Unit**, then click on OK.

Both variables have now been created, all that remains is to insert them into the document (see **Figure 22**).



Calculating an expiry date using the system date.

Date of manufacture: 25/08/00

EXPIRY DATE: 27/08/01

Figure 22 Calculating an expiry date

The label is now finished. The label stock is located in TUTORIAL\SYSTDATELAB.

Practical Workshop 4 – Date Calculations

Calculating an expiry date using a non-system date

In this section, we are going to reproduce the same document without using the system date. Let us suppose that the operator enters the date of manufacture just prior to printing the label. This method means that the expiry date must be calculated using the date of manufacture.

A number of formulas will thus be required to create the document. All calculations will need to be carried out using days as the unit of measurement.

The main functions used will be **day()**, **month()**, and **year()**. The method involves adding a time period to the day, month and year.

These three functions extract the current day, month and year if the argument is a system date, i.e. **today()**. However, they can also return the same result if the argument entered is the number of days elapsed since a reference date (see the example below).

Thus day(0)&”/”&month(0)&”/”&year(0) returns the result: **30/12/1899** = your application’s reference date.

If you wish the formula to return the current date, the solution is to calculate the number of days elapsed since the reference date.

The formula must contain the sum of:

▶ 99*365 = total number of days for 99 years of 365 days;
▶ quotient(99-1.4) = number of extra days for leap years;
▶ +1 = allows you to set the reference date at 31/12/1899 so as to start at a date minus one day;
▶ + the number of days elapsed during the current year, let us say that it is 20 January 1999.

The expression is thus:

day(99*365+quotient(99-1.4)+1+20)&”/”&month(99*365+quotient(99-1.4)+1+20)&”/”&year(99*365+quotient(99-1.4)+1+20)

and the result is **20/01/1999**, which is the current date we have taken for this example.

▶ **To create the intermediate variable required for entering the date of manufacture:**

We will start by creating the variables on which we will base our calculations.

1. Create a new document.
2. Add an intermediate variable (**Free** branch), and rename it **DATE**.
3. Configure the properties dialog box for this variable with the following values:
   - select the **Local** option;
   - enter ”28021999” as the date of manufacture in the box;
   - check the **Delete spaces** and **Display in Form** options;
   - enter **##/##/####** as the **Input format**.
   - enter **8** as **Minimum length**.
   - select the **Output** tab and enter ”8” as the **Maximum length**.
4. Click on OK.

The variable has disappeared from the **Free** branch! Checking the **Display in Form** option transforms the variable into a **Form** variable. Actually, all variables for which the data source is direct keyboard input are automatically transferred to the **Form** branch.

▶ **To create the Var variable for entering the shelf life of the product:**

- Repeat the previous steps, modifying the parameters as follows:
  - enter ”SHELF LIFE DAYS” as the variable name;
  - select the **Local** option;
  - enter **367** in the edit box as the value for the shelf life;
  - check the **Delete spaces** and **Display in Form** options.

▶ **To create the DAY, MONTH, and YEAR intermediate variables:**

These three variables are extracted from the **DATE** variable, i.e. the date of manufacture. As we shall see later, they will be required for calculating the expiry date.

1. In the **Document Browser**, create a **Formula** variable, and name it **DAY**.
2. Enter the following expression in the formula edit box: **left(DATE,2)**: this function extracts two characters from the **DATE** argument, starting from the left. Given that the current value of the **DATE** variable is ”28021999”, this formula will thus give as result of ”28”.
3. Click on OK.
4. Repeat the previous step to create the **MONTH Formula**, entering the following expression: **mid(DATE,3,2)**.
   This function returns two characters, starting from the third character in the **DATE** string, thus giving a result of **02**.

5 Click on OK.

6 Repeat the previous step to create the YEAR **Formula variable, entering the expression: right(DATE,4)-1900**. This function extracts four characters from the DATE argument, starting from the right, and thus giving a result of **1999**. Subtracting 1900 allows us to work with three digits instead of two, thus guaranteeing **Year 2000** compliance.

7 Click on OK.

Note

The result of this formula cannot be used directly. We will discuss how to employ it a little later on.

▶ **To create the intermediate variable for calculating the number of days elapsed during the current year:**

The last element to be determined is the number of days elapsed in the current year. This variable will allow us to calculate this number, also checking whether or not the current year is a leap year.

In order to create this variable, we must first create an intermediate variable which we will call DAYSTRING. This variable will include a character string defining the number of days in a year of 365 days. The trick is to add the number of days elapsed to each consecutive month, starting from 1 January (see below).

000	**031**	**059**	**090**	**120**	**151**	**181**	**212**	**243**	**273**	**304**	**334**
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov

1 Using the **Free** data source, create the intermediate variable DAYSTRING.

2 In the variable dialog box, enter the above string (000031059090120151181212243273304334) as the **Local** value, without including any spaces.

3 Click on OK.

The DAYSTRING variable will be used to calculate the number of days elapsed since 1 January of the current year. We now need to create the formula that will carry out this calculation.

4 Using the **Formula** tab, create a variable and name it JULIAN.

5 Enter the following expression into the edit box:
DAY+mid(DAYSTRING, (MONTH-1)*3+1.3)+if(and((MONTH > 2),(mod(YEAR,4)=0)),1,0)

DAY: corresponds to the date of manufacture (the variable we defined earlier).
mid(DAYSTRING,(MONTH-1)*3+1.3): the formula allowing us to extract from the **DAYSTRING** variable, the characters corresponding to the month preceding the month of the date of manufacture.
if(and((MONTH > 2) ,(mod(YEAR,4)=0)),1,0): allows us to check whether the current year is a leap year, in which case if the date of manufacture is after February, an extra day needs to be added.

Since the date of manufacture in our example is the 28021999, 31 + 28 = 59 days have passed since 1 January.

The **JULIAN** formula thus confirms this result.

▶ **To create the formula for calculating the expiry date:**

All the elements are now in place for defining the expression that will calculate the expiry date.

1 Create a formula, and name it **EXPIRY DATE**.

2 Enter the following expression into the edit box:
right(”00”&DAY(YEAR*365+quotient(YEAR-1.4)+1 +JULIAN + EXPIRY DAYS),2)&”/”& right(”00”&month(YEAR*365+quotient(YEAR-1.4) +1+JULIAN + EXPIRY DAYS),2) & ”/” & right(year(YEAR*365+quotient(YEAR-1.4)+1+JULIAN + EXPIRY DAYS),4)

As we have already seen in the introduction to this example, the functions that allow such calculations to be made are :
- **day**(”days elapsed in the last year -1” + ”product life span in days”) and;
- **month**(”days elapsed in the last year -1” + ”product life span in days”) and;
- **year**(”days elapsed in the last year -1” + ”product life span in days”).

In our example, the **EXPIRY DATE given will be 01/03/2000**.

Note The label we have just produced is also available as an example label in the TUTORIAL\LABELS\EXPRDATE.LAB folder.

Adding comments so that everything is clear.
When a number of different people are involved in developing a document, it can be very useful to include certain comments so that each user understands the reasons why a given calculation has been included. The entire area outside the borders of the label is non-printable, allowing you to insert, among other things, the necessary explanations and comments.

The Online Help provides full information and further examples of all the functions used in this example.

Appendix 2 – Connecting to Databases

A few reminders

This chapter should give you some idea of just how powerful your software really is. We are now going to link your label (the container) with a database (the content) using **ODBC** connections(**Open Data Base Connectivity**).

Database

Databases allow you to store data, which is organized into two-dimensional **tables** called a **relationship**. Each row in a table is called a **record**. The purpose of a record is to manage an object, the properties of which are organized across the different columns of the table in the form of **fields**.

A database can contain a number of tables. To link the different tables in a given database, we use **joins**. A concrete example later in the chapter demonstrates how joins are created.

ODBC

This is the database access standard. ODBC offers a straightforward method for linking an application, such as your label design software, to a number of different databases.

The software provides a number of ODBC drivers, offering you access to the very latest databases. The drivers are listed below:

- Microsoft Access Driver (*.mdb)
- Microsoft Excel Driver (*.xls)
- Microsoft FoxPro Driver (*.dbf)
- Plus a number of others.

To sum up...

What is a database?

A database allows you to store data in the form of tables. A table contains records that allow you to manage objects, the properties of which are organized across a number of fields.

How do I access a database?

By using ODBC.

Where does the data come from?

From an ODBC data source. The data source includes all the necessary information for accessing the database, such as its name and access path, and the type of database engine associated.

Practical Workshop 1– Installing an ODBC Data Source and Importing Data

Before the data can be accessed, the first step involves installing the necessary data source.

Installing the ODBC data source

This first example demonstrates how to install a data source for an Access™ database.

▶ **To install the data source:**

1 Open the RECTO_00.LAB label located in the product installation folder \TUTORIAL\LABELS.

It is a sleeve for a CD containing the title of the work and the tracks recorded.
The data to be inserted into the label is located in the database entitled VOLMUSIC.MDB.

2 Choose **Merge Create/Edit query...**

The **ODBC query definition** dialog box appears.

3 Click on ... to the right of the **Select a data source** list.

The **ODBC data source administrator** dialog box appears.

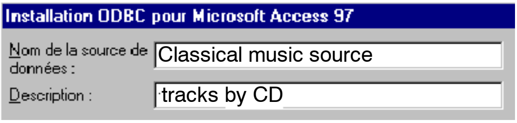
ODBC is a component completely separate from your label design software. You can also access the **ODBC Data Source Administrator** via the Windows control panel: choose **Start Settings Control panel** and click on the icon below:

ODBC

4 In the **User DSN** tab, click on **Add...**, then select the **Microsoft Access Driver (*.mdb)**. Click on **Finish**.

The **ODBC installation for Microsoft Access** dialog box appears.

5 Enter a prompt in the **Data source name** box, for example **Classical music source** and, if need be, give a brief description (see diagram below)



Installation ODBC pour Microsoft Access 97

Nom de la source de données : Classical music source

Description : tracks by CD

Figure 23 To name a data source

A data source is now associated with a database. The next step involves selecting the database we need for our example, i.e. VOLMUSIC.MDB.

▶ **To select the database:**

1 Still in the **ODBC installation for Microsoft Access** dialog box, click on **Select** in the **Database** zone.

2 Select the database located in the installation folder of your label design software \TUTORIAL\DATA\VOLMUSIC.MDB, then click on OK.

3 Close the **ODBC installation for Microsoft Access** dialog box by clicking on **OK**, then exit the **ODBC data source administrator**, again clicking on **OK**.

Online Help is available for each **ODBC driver**, do not hesitate to consult it!

Importing data

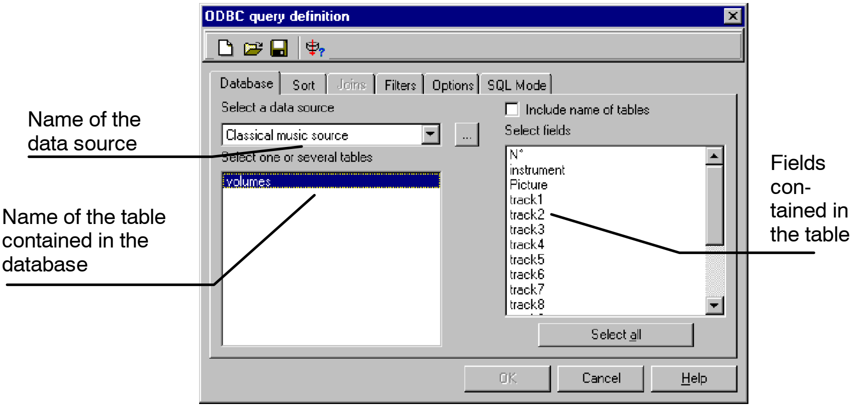
Now that the data source has been installed, we can import the data from the database and insert it into the label.

▶ **To import the data:**

1 If the **ODBC query definition** dialog box is not displayed on the screen, choose **Merge Create/Edit query...**

2 Choose **Classical music source** from the **Select a data source** list, then select the **volumes** table.

The dialog box displays the following information:



Select a data source: Classical music source

Select one or several tables: volumes

☐ Include name of tables

Select fields: N°, instrument, Picture, track1, track2, track3, track4, track5, track6, track7, track8

Figure 24 The **ODBC query definition** dialog box

3 Click on **Select all**, then on OK.

All **Database** variables will now be created automatically from the fields in the database.

The variables are listed within the **Database** branch in the **Document Browser Data Sources** tab.

4 Save your document.

The application asks you if you wish to save the query.

5 Click on **Yes**, select \TUTORIAL\QUERY and enter VOLMUSIC.CSQ as the query name.

Creating variable objects

All that remains to be done is to associate the variables we have created with an object (text, barcode, etc.) so that they can appear in the label. For the purposes of our example, we are going to create variable **Text Objects**.

Reminder! It is important to distinguish between **variables** and **variable objects**. A **variable** is a non-physical object, the value of which changes. A **variable object** has a physical appearance (text, barcode, etc.).
When an object (text, barcode, etc) is associated with a variable, it is called a **variable object**.
A given **variable** can take on different physical appearances: it can be associated to both a **Text** object and a **Barcode** object at the same time.
Moreover, you can create as many variables as you wish without creating variable objects. Such variables can be useful for creating formulas, for example. We will look at this later on in the chapter (see Chapter 8 - 77, **To concatenate variables**).

▶ To create the variable text object ”Instrument”:

1 In the label, select the fixed text ”Title” (the whole text must be highlighted).

2 Choose **Object properties...** in the context menu.

3 Click on the **Data source** tab.

4 Check the **Variable data** option, then select the **Instrument** field located within the **Database (18)** branch.

5 Click on OK.

You have just changed a fixed object into a variable object.

- To view the different values that the object can take, use the **Navigation bar**.

You will notice that the variable object takes the following values respectively: “The Oboe”, “The Flute”, “The Horn”, “The Harpsichord”, “The Piano”, “The Trumpet”, “The Organ”, “The Cello”, “The Guitar” and “The Violin”.

Do not forget to click on the **Display field contents** button to view the variable values.

▶ **To create the variable image ”Picture”:**

1 Select **Picture** located within the **Database** branch of the **Document Browser**.

2 Drag and drop **Picture** into the document.

A context menu appears.

3 Choose **Image**.

The image appears on the screen.

4 Select it, then choose **Properties...** in the context menu.

5 Click on the **Options** tab.

6 Check the **Other folder** option and select the folder .\TUTORIAL\IMAGES, then click on OK in the **Image** dialog box.

7 Select the image and position it towards the top of the document. Readjust its size if necessary.

▶ **To create the variable object** “**N°**”**:**

1 Select the **Text generation** tool, then select the **N°** variable located within the **Document Browser Database** branch.

2 Keeping the variable selected, drag it into the label and drop it next to the ”Volume:” text.

3 Save your document.

▶ **To concatenate variables:**

In order to finish the sleeve we need to create the variable objects by associating them with the variables **Track1** through **Track15**.

Using the method outlined above, associating each of the 15 variables to an object, thus creating 15 variable objects.

To save time, and to facilitate data management later, we will just create one variable object containing all the data we need.

To do this, we are going to use the **Formula** variable, creating an expression which is the concatenation of all variables: **Track1** through **Track15**.

1 Select **Formula** in the **Document Browser Data Sources** tab, then choose **Add...** in the context menu.

A variable **Formula0** is automatically created.

2 Select this sub-branch, then double-click on it.

The **Formula** dialog box appears.

3 Select **Variables** from the list of different elements that can be included in the formula.

You will notice that **Variables** contains all the variables available in you application for the current label:
- the application control variables (see **Chapter 12 - 123**);
- plus the variables you have just created by merging the document with the VOLMUSIC.MDB database.

We now need to concatenate the variables **Track1** through **Track15**. In order to obtain an organized display, we will also insert text strings containing the track number followed by a period and a space into the formula.
(example: “1. “).

4 Enter the expression: ”1. ”& track1 & ” - 2. ”& track2 & ” - 3.”&track3 &” - 4. ”& track4 & ” - 5. ”& track5 & ” - 6. ”& track6 & ” - 7. ”& track7 & ” - 8. ”& track8 & ” - 9. ”& track9 & ” - 10. ”& track10 & ” - 11. ”& track11 &” - 12. ”& track12 & ” - 13. ”& track13 &” - 14. ”& track14 & ” - 15. ”& track15

5 Click on OK.

6 Name it ”Tracks”.

We now need to associate the **Tracks** formula variable to a **Text** object so that it can appear in the label.

▶ **To create a variable Text object:**

1 Select the **Text generation** tool, then the **Tracks** formula variable.

2 Drag the formula into the label and drop it below the **Instrument** variable.

If you wish to view the content of the variable, do not forget to select the **Content** view mode (see Chapter 8 - 76, **Display field contents**).
Moreover, if the variable value is truncated, check the display settings in the **Output** tab of the variable's dialog box. Enter **255** in the **Maximum length** box.
Finally, to make a paragraph of several lines, check the **Word-wrap** option (**Paragraph** tab of the **Text** dialog box) and adjust the size so that it fits nicely into the lower half of the sleeve.

3 Save your document.

Creating the Table lookup variable

The **Table lookup** variable allows you to search for specific data in a database or table other than the one merged with the current label.

It may be, for example, that certain object properties are stored in another database. Storing data in different databases or tables makes them lighter and easier to work with, avoiding data repetition.

In our example, it happens that the VOLMUSIC.MDB database only contains part of the data we need. In order to complete the sleeve, we need the following information: Conducted by:.

This information is stored in a second Access™ database called CNDTEDBY.MDB. This database contains the name of the conductor who directed the recording of each volume.

▶ **To create the Table lookup variable:**

1 Select **Table lookup** in the **Document Browser Data Sources** tab, then click on it with the right mouse button and choose **Add...**

Within the **Table lookup** branch, a sub-branch is automatically created specifying the name of the data source.
In this way, as many sub-branches are created as there are **Table lookup** variables with result fields originating from different databases.

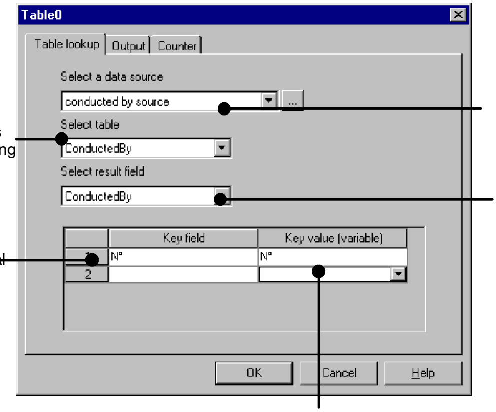
2 Double-click on the sub-branch.

The **Table lookup** dialog box appears on the screen.

3 Connect the CNDTEDBY.MDB database using the **ODBC Data Source Administrator** (see Chapter 8 - 73).

4 Once the connection has been made, select the name of the data source in the **Select a data source** list in the **Table lookup** dialog box.

5 Complete the different fields as shown in **Figure 25** below.



Select a data source: conducted by source

Select table: ConductedBy

Select result field: ConductedBy

	Key field	Key value (variable)
1	N°	N°
2		

Allows you to select the table that contains the data you are looking for.

Allows you to search for
the data by selecting the field in the external table in which the search will be carried out.

Allows you to connect to another data source.

Allows you to select the field in the table that contains the data you are looking for.

Lists all the variables in the current label.
Select the one with the value that will allow the result field to be displayed.
In our example, the key value is the volume number: **N°**. This information is also stored in the CNDTEDBY.MDB database and corresponds to the name of the conductor.

Figure 25 Creating a **Table lookup** variable

6 Once the dialog box has been completed, click on OK.

7 Select the **Table lookup** variable that you have just created, then click on it once to select the default name and enter **Name**.

8. Select the **Text generation** tool, then, keeping the **Name Table lookup** variable selected, drag and drop it under the ”Conducted by” text.
9. Save your document.

To sum up...

What is a variable ?

A variable is a non-physical object, the value of which advances as printing progresses. There are different types of variable. The value of a variable depends on the type of data source.

What are the different types of variable?

There are eight different types:

- **Database**
- **Table lookup**
- **Formula**
- **Counter**
- **Date**
- **Form**
- **Free**

Which data sources can feed these variable?

- The **Database** variable is fed by the data contained in a field of the database merged with the current label.
- The **Table lookup** variable is fed by the data contained in a field of a data source other than the one merged with the current label.
- The **Formula** variable is fed by the result of an expression that can contain variables, constants, and so on (see Chapter 7 - 53, **Appendix 1 – Formulas**).
- The **Counter** variable is fed by the result of a calculation carried out by the computer or the printer.
- The **Date** variable is fed by the system date of your computer or printer.
- The **Form** variable is fed by keyboard input.

What is a variable object?

A variable object is a variable that takes on a physical appearance, whether text, an image or a barcode.

Practical Workshop 2 – Complex Queries and Joins

In this next sequence, we are going to look at the concept of queries and joins. To do this, we are going to create the reverse side of the CD sleeve containing the volume number, artist’s name, tracks list, and track times.

To create the reverse side of the sleeve, we are going to use a third Access™ database that holds the supplementary information we need, i.e. the track times and the artist. This database is called VOLDUR.MDB and contains two tables:

- **Duration**
- **Author**

Constructing an internal join

The first step involves connecting the new label VERSO_00.LAB to the database VOLDUR.MDB.

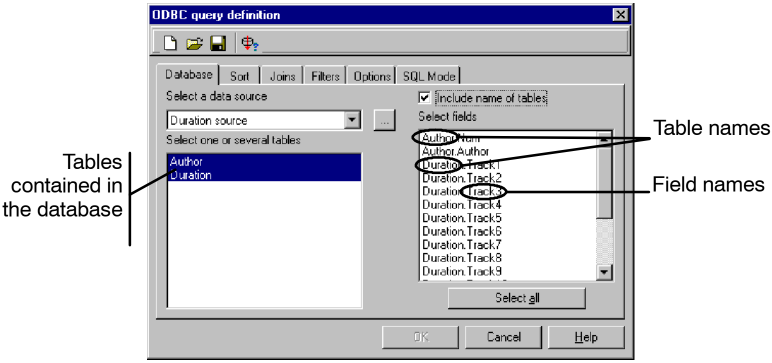
▶ **To create the data source:**

1 Open the label VERSO_00.LAB, located in the TUTORIAL\LABELS folder, then choose **Merge Create/Edit query...**

2 Create a new data source using the **ODBC data source administrator**. If necessary, refer back to Chapter 8 - 73.

3 Name the new source ”Duration source”.

4 Once the new source is installed, select it in the **Select a data source** list in the **ODBC query definition** dialog box (see **Figure 26** below).



Select a data source: Duration source

Select one or several tables: Author, Duration

☑ Include name of tables

Select fields:
Author.Num
Author.Author
Duration.Track1
Duration.Track2
Duration.Track3
Duration.Track4
Duration.Track5
Duration.Track6
Duration.Track7
Duration.Track8
Duration.Track9

Figure 26 Selecting the data source **Duration source**

5 Select the two tables: **Author** and **Duration**.

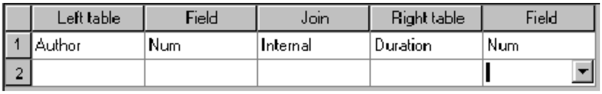
All the fields contained in the two tables appear in the list on the right hand side.

6 Check the **Include name of tables** option.

This option displays the source of a given field.

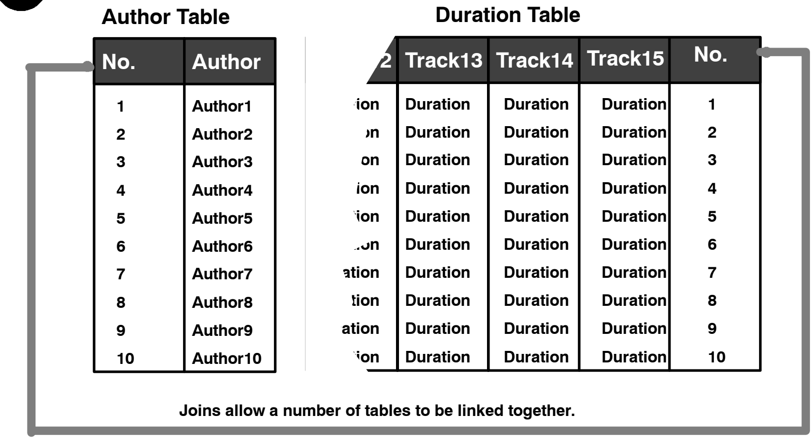
7 Click on **Select all**, then on the **Joins** tab.

8 Enter the expression for defining the join as shown in **Figure 27** below.



	Left table	Field	Join	Right table	Field
1	Author	Num	Internal	Duration	Num
2					

1



Author Table

No.	Author
1	Author1
2	Author2
3	Author3
4	Author4
5	Author5
6	Author6
7	Author7
8	Author8
9	Author9
10	Author10

Duration Table

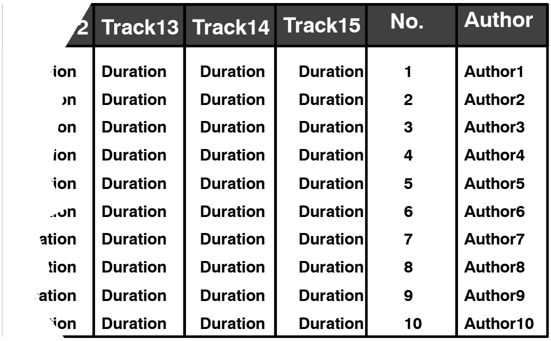
2	Track13	Track14	Track15	No.
ion	Duration	Duration	Duration	1
on	Duration	Duration	Duration	2
on	Duration	Duration	Duration	3
ion	Duration	Duration	Duration	4
ion	Duration	Duration	Duration	5
on	Duration	Duration	Duration	6
ation	Duration	Duration	Duration	7
tion	Duration	Duration	Duration	8
ation	Duration	Duration	Duration	9
ion	Duration	Duration	Duration	10

Joins allow a number of tables to be linked together.

Creating a join linking two tables allows you to obtain the data contained in the **Author** table and that in the **Duration** table at the same time.

2 To display the result of the query, click on

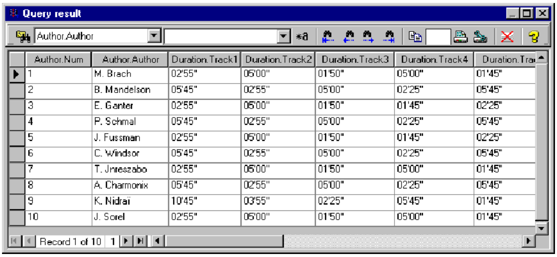
Query result



2	Track13	Track14	Track15	No.	Author
ion	Duration	Duration	Duration	1	Author1
on	Duration	Duration	Duration	2	Author2
on	Duration	Duration	Duration	3	Author3
ion	Duration	Duration	Duration	4	Author4
ion	Duration	Duration	Duration	5	Author5
on	Duration	Duration	Duration	6	Author6
ation	Duration	Duration	Duration	7	Author7
tion	Duration	Duration	Duration	8	Author8
ation	Duration	Duration	Duration	9	Author9
ion	Duration	Duration	Duration	10	Author10

Figure 27 Constructing an internal join

9 Once you have displayed the result of the query (see below), close the **Query result** grid, then click on OK in the **ODBC query definition** dialog box.



	Author.Num	Author.Author	Duration.Track1	Duration.Track2	Duration.Track3	Duration.Track4	Duration.Tra
▶	1	M. Brach	02'55"	05'00"	01'50"	05'00"	01'45"
	2	B. Mandelson	05'45"	02'55"	05'00"	02'25"	05'45"
	3	E. Ganter	02'55"	05'00"	01'50"	01'45"	02'25"
	4	P. Schmal	05'45"	02'55"	05'00"	02'25"	05'45"
	5	J. Fussman	02'55"	05'00"	01'50"	01'45"	02'25"
	6	C. Windsor	05'45"	02'55"	05'00"	02'25"	05'45"
	7	T. Jnreszabo	02'55"	05'00"	01'50"	05'00"	01'45"
	8	A. Charmonix	05'45"	02'55"	05'00"	02'25"	05'45"
	9	K. Nidraï	10'45"	03'55"	02'25"	05'45"	01'45"
	10	J. Sorel	02'55"	05'00"	01'50"	05'00"	01'45"

Figure 28 The **Query result** grid

The **Database** branch now displays the 18 variables created.

10 Save your document and the query, naming it, for example, VOLDUR.CSQ.

Inserting variables resulting from a query

In this next sequence, we are going to insert the variables into the label. We could associate each track length variable with a **Text** object, but that would be time-consuming and laborious. As in the previous exercise, we are going to create a formula concatenating the **Duration** variables.

▶ **To concatenate the variables:**

1 Select **Formula** in the **Document Browser Data Sources** tab, then choose **Add...** in the context menu.

A **Formula0** variable is automatically created.

2 Select the sub-branch, then double-click on it.

3 Select **Variables** from the list of different elements that can be included in the formula.

4 Enter the following expression:[1] : "Duration.Track1&"\n"& Duration.Track2&"\n" &Duration.Track3&"\n"& Duration.Track4&"\n"& Duration. Track5&"\n"& Duration. Track6&"\n"& Duration.Track7&"\n"& Duration. Track8&"\n"& Duration.Track9&"\n"& Duration.Track10&"\n"& Duration.Track11&"\n"& Duration. Track12&"\n"& Duration.Track13&"\n"& Duration.Track14&"\n"& Duration. Track15"

5 Name the variable "TrackDuration".

[1] \n = carriage return

We now need to associate the **TrackDuration** formula variable with a **Text** object so that it can appear in the label.

▶ **To create a variable Text object:**

1 Select the **Text generation** tool, then, keeping the **TrackDuration** formula variable selected, drag and drop it into the document.

2 Save your document.

If the value of the variable is truncated, check the output settings. The maximum length probably needs adjusting.

Inserting the titles using the Table lookup variable

To complete the reverse side of the sleeve, we now need to insert the titles of the different tracks opposite their respective track lengths.

▶ **To insert the titles:**

1 Create a **Table lookup** variable in the **Document Browser**, and name it **Track1**.

2 Complete the dialog box with the following data:
- **Select a data source**: Classical music source;
- **Select table**: Volumes;
- **Select result field**: Track1;
- **Key field**: N;
- **Key value (variable)**: Author.Num.

3 Repeat the above procedure to create all 15 **Table lookup** variables, not forgetting to select the corresponding **Result field**: Track1 for the Track1 variable, Track2 for the Track2 variable, and so on.

You will by now have noticed that the **Table lookup** variables are classified according to their source. A sub-branch is automatically created and is named after the table the data was extracted from. When you need to create a number of variables using data from the same source, add them directly to the corresponding sub-branch. In this way, the properties dialog box will be pre-configured—you will not need to specify the data source or the table each time.

4 Once all the variables are created, save your document.

▶ **To create the concatenation formula**

In the same way as for the data for the track lengths, we are going to create a formula associating the data for the track titles by inserting the carriage return function to produce the same display format as last time.

1 Create a second formula and name it ”TrackTitle”.

2 Enter the expression:
Track1&”\n”&Track2&”\n” &Track3&”\n”& Track4&”\n”&
Track5&”\n”& Track6&”\n”&Track7&”\n”& Track8&”\n”&
Track9&”\n”&Track10&”\n”& Track11&”\n”&Track12&”\n”&
Track13&”\n”&Track14&”\n”& Track15”

3 Click on OK.

4 Select the **Text generation** tool, then drag and drop the **TrackTitle** formula next to the **TrackDuration** formula.

Use the alignment tools to correctly position the data within your document.
Select the two formulas and choose **Object Alignment Top**.
Once the data is perfectly aligned, lock its position: **Object Lock / Unlock**.
If the data does not fit within your document, you will need to modify the text properties. For example, choose **Arial 10** point.

▶ To finish off the document

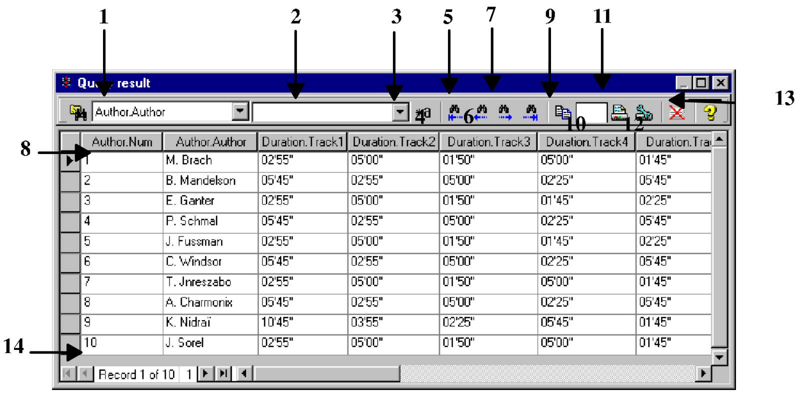
All that remains now is to insert the volume number and artist.

1 Select the **Text generation** tool, then the **Author.Num Database** variable. Drag and drop it next to the ”Volume:” text.

2 Repeat the above operation to position the artist’s name next to the ”Author:” text.

Practical Workshop 3 – The Query Result Grid

The **Query result** grid allows you to view the result of a query, search for a particular term or all occurrences, and print the required records (see **Figure 29** below).



Author.Author

	Author.Num	Author.Author	Duration.Track1	Duration.Track2	Duration.Track3	Duration.Track4	Duration.Tra
▶	1	M. Brach	02'55"	05'00"	01'50"	05'00"	01'45"
	2	B. Mandelson	05'45"	02'55"	05'00"	02'25"	05'45"
	3	E. Ganter	02'55"	05'00"	01'50"	01'45"	02'25"
	4	P. Schmal	05'45"	02'55"	05'00"	02'25"	05'45"
	5	J. Fussman	02'55"	05'00"	01'50"	01'45"	02'25"
	6	C. Windsor	05'45"	02'55"	05'00"	02'25"	05'45"
	7	T. Jnreszabo	02'55"	05'00"	01'50"	05'00"	01'45"
	8	A. Charmonix	05'45"	02'55"	05'00"	02'25"	05'45"
	9	K. Nidraï	10'45"	03'55"	02'25"	05'45"	01'45"
	10	J. Sorel	02'55"	05'00"	01'50"	05'00"	01'45"

Record 1 of 10 1

<u>Search functions</u>:

1 *Search field*

2 *Search value*

3 *Search any part of the data field*

<u>Moving between occurrences of the search results</u> :

4 *First occurrence*

5 *Previous occurrence*

6 *Next occurrence*

7 *Last occurrence*

<u>Printing</u>:

8 *Enter the number of labels in the series to be printed.*

9 *Select the row(s) you wish to print, and enter the number of label copies to be printed for these record(s).*

10 *Print*

11 *Printer setup*

12 *Reset all print quantities defined in the first column*

13 *Help*

14 *Information bar*

Figure 29 The **Query Result** Grid

Appendix 3 – Counters

Numbering your documents

The **Counter** variable is integrated into your label design software. The flexibility of the counter variable makes it suitable for a number of different applications such as counting a series of labels or totalling the number of labels printing in a day.

You can use this variable to create ascending, descending, linked and shared counters.

This section is complemented with numerous examples covering many of the options available when working with counters. The examples entitled **Practical Workshop** are set at different levels of difficulty, the level indicated by one of the following icons:

☺	**Very easy**	☺	**Easy**
☺	**Difficult**		

Creating a counter

The content of a **Counter** variable advances from one label to the next during printing, according to an initial value and an increment.

The **Counter** variable dialog box is used to define the attributes of the **Counter** variable.

▶ **To create a counter:**

1 Select **Counter** in the **Document Browser Data Sources** tab.

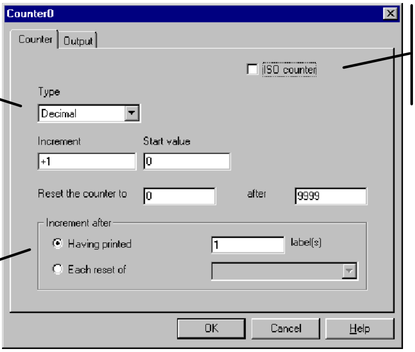
2 Choose **Add...** in the context menu.

3 Double-click on the counter thus created.

The Counter **dialog box** appears (see **Figure 30** below).

offers a list of six counting bases, plus a **Custom** option: Binary, Octal, Decimal, Hexadecimal, Alphabetical, Alphanumeric.

allows the event that triggers the counter increment to be determined.



□ ISO counter

Type: Decimal

Increment: +1

Start value: 0

Reset the counter to 0 after 9999

Increment after

● Having printed 1 label(s)

○ Each reset of

allows each label copy to be individually and exhaustively numbered.

Figure 30 The **Counter** dialog box

The dialog box comprises two tabs:

The **Counter** tab allows you to define the type of counter and its characteristics (see **Online Help, Counter Tab**);

The **Output** tab allows you to format the final output of the variable for display in your document (see **Online Help, Output Tab**).

Practical Workshop 1 – Counter for a Series of Labels ☺

In this sequence, we are going to create a label containing a counter dedicated to counting labels in a series. The number of labels in the series is fixed.

Numbering labels in a series

▶ **To create a numbering system for a label series:**

1 Open a new document and name it SERIALCT.LAB.

2 Select **Counter** in the **Document Browser Data Sources** tab.

3 Choose **Add...**in the context menu.

The **Counter** variable dialog box appears (see **Figure 30**).

4 Complete the dialog box with the following values:
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 1;
- **Reset the counter to**: 1;
- **After**: 10.

5 Click on the **Output** tab and enter ”Label” as the prefix.

6 Click on OK and name the variable **LABELNUM**.

7 Select the **Text generation** tool then drag and drop the **LABELNUM** variable into the lower right-hand corner of your document.

Do not forget to select the **Content** display mode in order to view the counter value.

You have now finished creating a counter that will number labels in a series. For the sake of our example, let us suppose that the series contains ten labels.

Total number of labels in the series

We now need to create a variable that will display the total number of labels in the series for each label in the series. To do this we are going to use one the application control variables.

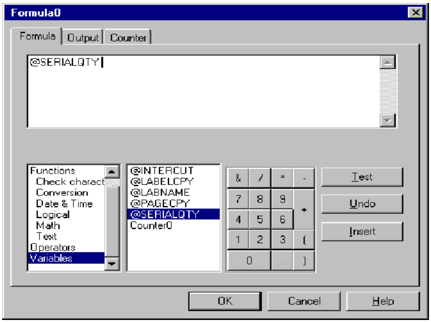
▶ **To insert the total number of labels:**

1 Select **Formula** in the **Document Browser Data Sources** tab.

2 Choose **Add...** in the context menu.

3 Double-click on the formula thus created.

4 Insert the **@SERIALQTY** control variable, situated under **Variables** (see **Figure 31** below).



@SERIALQTY

Figure 31 Using a control variable in a formula

5 Click on OK and name the formula **TOTALLABEL**.

6 Select the **Text generation** tool, then drag and drop the formula next to the **LABELNUM** counter.

You will notice that the value of **LABELNUM** is 1 while it should display 10, since the series contains 10 labels. This is because the variable will not be updated until printing starts. Setting the number of labels in the series is carried out using the **Print** dialog box.

▶ **To print:**

1 Choose **File Print...** or click on or press the **F6** key.

2 Enter **10** in the **Labels in the series** box, then launch printing.

Note

This label has been created for an office printer using a Windows™ driver. The page setup has been configured accordingly :
- format: A4;
- label size: 80/50;
- per row: 2;
- per column: 5.
The page has been deliberately set up to display all ten labels on the same page.

Practical Workshop 2 – Customized counter ☺

In this sequence we are going to create a label containing a counter, the counting base of which will be set by the user.

Creating a customized counter

▶ **To create a customized counter:**

1 Open a new document and create a **Counter** variable.

2 Complete the **Counter** variable dialog box with the following values:
-**Type**: Custom;
- **Customized sequence**: A1B2C3D4E5;
Start value: A;
Reset the counter to: A;
- **After**: 5.

3 Click on OK and name the counter ”Customize”.

4 Select the **Text generation** tool, then drag and drop the counter into your document.

The customized sequence must not contain repeated characters or spaces.

▶ **To print:**

1 Choose **File Print...** or click on the 🖨 button or press the **F6** key.

2 Enter **10** in the **Labels in the series** box, then launch printing.

The result obtained will be as follows:

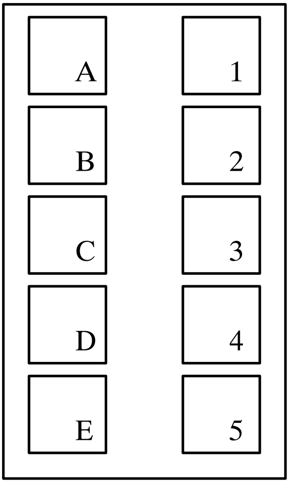
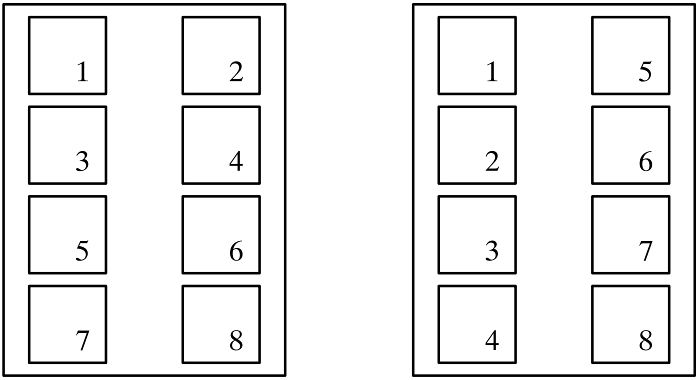


Figure 32 Customized counter

Practical Workshop 3 – Linked Counters

By combining several counters, you can create totally customized numbering systems. In this sequence you are going to create a label numbering system different to the one offered by default. Instead of numbering labels from left to right (across the row of labels contained in the page), numbering will be carried out from bottom to top, i.e. up the column of labels contained in the page (see **Figure 33** below).



Classical counter: numbering from left to right.

Linked counter: numbering from top to bottom.

Figure 33 Example of complex numbering

Creating a combined counter

The trick here is to distinguish between the left and right columns. One counter must be created to number the left column, a second to number the right column. Conditions are verified by a third counter, which is binary. If necessary, a fourth and final counter may be employed to number the following pages.

▶ **To create intermediate counters:**

1 Open the TOPTOLOW.LAB label, located in the TUTORIAL\LABELS folder of your application.

2 Using the **Document Browser**, add a counter and name it **LEFT**. Complete the dialog box with the following values:
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 1;
- **Reset the counter to**: 1 **after** 4;
- **Increment after having printed** 2 **labels**.

3 Add a second counter and name it **RIGHT**. Complete the dialog box with the following values:
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 5;
- **Reset the counter to**: 5 **after** 8;
- **Increment after having printed** 2 **label(s)**.

4 Add a third counter and name it PAGE. Complete the dialog box with the following values:
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 0;
- **Reset the counter to**: 0 **after** 9999;
- **Increment after having printed** 8 **label(s)**.

5 Add a fourth counter and name it **TEST**. Complete the dialog box with the following values:
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 1;
- **Reset the counter to**: 1 **after** 2;
- **Increment after having printed** 1 **label(s)**.

All the counters required for the numbering system have now been created. We now need to combine them in a formula.

▶ **To combine counters:**

1 Using the **Document Browser**, add a formula and name it NUMBER. Enter the following expression:
if (TEST = 1, LEFT, RIGHT) + PAGE*8

2 Insert the NUMBER formula into your label.

3 Choose **File Print...** or click on the button or press the F6 key.

4 Enter "16" in the **Labels in the series** box, then launch printing.

The printed document should look like this:

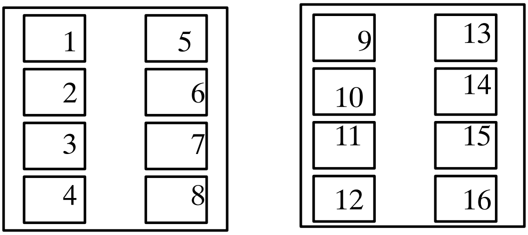


Figure 34 Printed document

Practical Workshop 4 – Determining the Total Number of Labels in a Subseries 😐

Let us suppose that you wish to print a completely random number of labels that must be classed into subseries, each containing a maximum of 40 labels. Given that the number of labels is random, the counter must also be able to determine a subseries of less than 40 labels and, if need be, to determine the total number of labels in this subseries.

For example, if you print 63 labels the first 40 will be numbered up to 40, the last 23 will make up a series of 23 and will be numbered as such.

Creating the label counter

1 Open a new document and name it PACK.LAB.

2 Using the **Document Browser**, select **Free**. Add a variable and name it **LABELNUM**. Complete the dialog box with the following values:
Input tab:
- **Local**: 1;
Restore after printing: yes.
Counter tab:
- **Counter**: check;
- **Type**: Decimal;
- **Increment**: +1;
- **Start value**: 1;
- **Reset the counter to**: 1 **after** 40;
- **Increment after having printed** 1 **label(s)**.

3 Select the **Text generation** tool, then drag and drop the **LABELNUM** counter into the lower left-hand corner of your document.

Why use a Free variable as a Counter?
At first sight, it would seem logical to create a counter. However, counter properties are limited and, in particular, do not offer a **Restore after printing** option. In our example, it is important to include this function so that we can launch successive print jobs without having to manually reset the counters to their initial values.

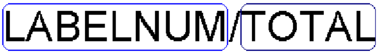
4 Repeating the preceding operations, create a second counter using the **Free** data source, and name it **SUBRANGE**.

5 Complete the dialog box with the following information:
Input tab:
- **Local**: 0;
Restore after printing: yes.
Counter tab:
- **Counter**: check;
- **Type**: Decimal;
- **Increment**: +40;
- **Start value**: 0;
- **Reset the counter to**: 1 **after** 9999;
- **Increment after having printed** 40 **label(s)**.

6 Using the **Formula** branch of the **Document Browser**, add a variable and name it **TOTAL**. Complete the box with the following expression (see comments Chapter 9 - 99):
if(value(@SERIALQTY- **SUBRANGE**)>value(40), 40 ,if (value (@SERIALQTY - **SUBRANGE**) = 0, 40,mod(@SERIALQTY ,40))).

7 Select the **Text generation** tool, then drag and drop the **TOTAL** formula into the lower left-hand corner of your document.

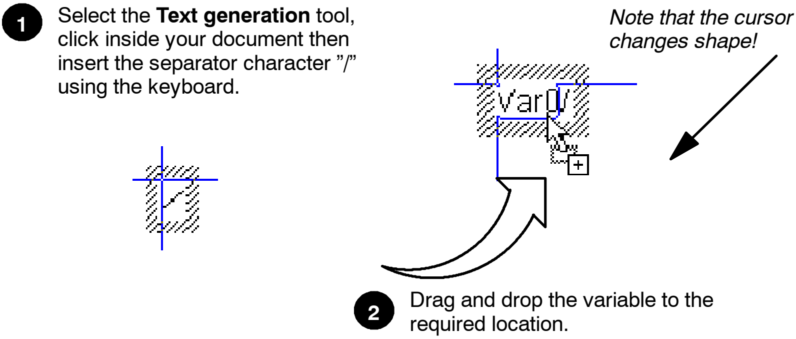
8 Select the **Oblique line drawing** tool and draw a separator as shown below:



LABELNUM/TOTAL

Run a test of 136 labels. The counter should display three subseries of 40 and one of 16.

More advanced!
There is a little trick enabling you to concatenate variables or fixed characters in the blink of an eye:



1 Select the **Text generation** tool, click inside your document then insert the separator character "/" using the keyboard.

Note that the cursor changes shape!

2 Drag and drop the variable to the required location.

Figure 35 To concatenate variables inside a **Text object**

Explanation of the formula:

if(value(@SERIALQTY- **SUBRANGE**)>value(40), 40 ,if (value (@SERIALQTY - **SUBRANGE**) = 0, 40,mod(@SERIALQTY ,40)))

For example, for the 63 labels to be printed we have:
- **from labels 1 to 40**: 63 - 0 = 63. Since 63 is greater than 40, the result is 40. The first 40 labels are thus numbered up to 40.
- **from labels 41 to 63**: 63 - 40 = 23. Since 23 is less than 40, the second condition must therefore be confirmed: Does 63 - 40 equal 0? 63 - 40 is not equal to 0, therefore the value to be displayed is the result of the subtraction 63 - 40, i.e. 23. Therefore, labels 41 to 63 are numbered up to 23.

For a series of 63 labels to be printed, we thus have one subseries of 40 labels and one of 23.

Appendix 4 – ActiveX Programming

10

Why get involved with programming?

There are several advantages to application programming in an automatic identification environment.

To make the most effective use of your label design software, routine and repetitive tasks can be automated.

The first section of this appendix provides a general overview of ActiveX, and a definition of the programming interface.

The second section gives an example of **Visual Basic 6.0** programming, introducing a number of functions that allow you to run your label design software from the programming interface.

ActiveX

To meet the need for high-performance integration and maximum flexibility, your software incorporates **ActiveX** technology.

ActiveX creates a customized environment for working between different applications, offering greater ease of integration and programming. Programmable objects can be accessed using any macro language or programming tool. Thus, using the application to run your label design software with a programming language such as Visual Basic is quite simple.

In short, **ActiveX** is a powerful tool allowing you to use your label design software as a label publishing and print engine. It is the cornerstone in the construction of personalized identification systems.

The programming interface

Functions

Carrying out a range of operations from another application is thus quite simple. For example:

- Load a label
- Obtain the list of variable fields
- Create variables
- Obtain the data from these fields
- Select the printer/port combination
- Access printer settings
- Access the **Add printer** dialog box
- Access the **Form**
- Save a label
- Update variable field values
- Obtain an image preview
- Print a document
- Create a complete label
- Plus a number of other functions

Using this technology you can develop your own interface and adapt it to the specific requirements of your organization:

- By monitoring the label printing process.
- By creating complex labels involving non-decimal counters or chronology tests automatically, for example.

Objects, attributes and methods

Any element that can be defined is known as an **Object**. Each object has associated attributes and methods.

Attributes describe the status of an object. The group of attributes together provides a complete and precise description of the object.

Attributes can be likened to adjectives in a sentence; methods to verbs describing an action.

The car **object**

- **Attributes** : color/year/make/number of seats/top speed.
- **Methods** : drive off/accelerate/break/stop.

Object hierarchy

The *Programmer's Guide* includes a diagram showing the hierarchical relationship between the various objects available in your label design software.

For example, in Visual Basic, the function " CreateObject(server name) " is all that is required to access the server. The period " . " is then used as a separator to access objects lower in the hierarchy (refer to the examples in the electronic appendix **ActiveX**).

Note

To find out the server name of your software, refer to the **About** dialog box in the **Help** menu. You can also use the generic name **Lppx2.Application**.
For detailed technical information on data types, refer to the *Programmer's Guide* available in the electronic appendix on the application CD-ROM. For further information on ActiveX, refer to the Microsoft documentation.

In the next sequence, we are going to look at an example of **Visual Basic 6.0** programming, highlighting the different objects and some of the associated methods and attributes.

Practical Workshop – Example of ActiveX

The program is simple to set up, and covers all the basic **ActiveX** functions. For a full description of the interface with the objects and associated methods and attributes, refer to the *Programmer's Guide*, available in the electronic appendix on the application CD-ROM.

Presentation of the application

The application uses **ActiveX** to drive your label design software. The program sources are available in the project TUTORIAL\ACTIVEX\FRSTPRGM.VBP.

The application allows you to open a document, create a document preview, and, if necessary, save the document in another folder or under another name. It also allows you to open a query, provided of course that the label is connected to a database, and to list the label variables. Moreover, it allows you to print all records, or individual ones. The navigation bar allows you to browse the database easily. You can also use the interface to display the client application, in this case your label design software.

Moreover, your label design software's **ActiveX** component uses a type library called **LabelManager2** listed as **TK Labeling ActiveX 6.0**. It contains all methods and properties associated with the objects you have access to.

▶ **To install the type library:**

1 Choose **Project References**.

2 Activate **TK Labeling ActiveX 6.0** in the list of available references, then click on OK.

▶ **To display the methods and properties associated with the objects:**

- Use the object browser in the Visual Basic programming interface (F2 key).

▶ **To display the code associated with a control:**

- Double-click on the control.

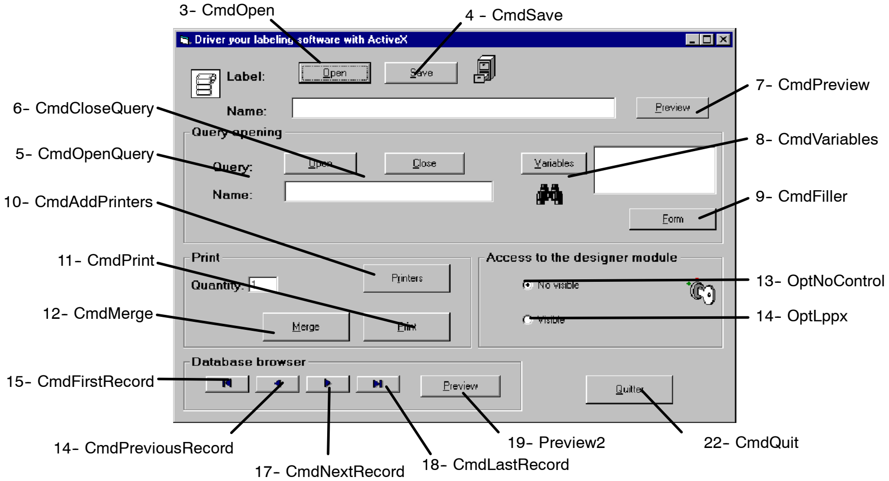


Figure 36 Presentation of the FRSTPRGM.VDP Interface

Note

This application illustrates the basic ActiveX functions, which are described in this chapter. For more information on ActiveX, refer to the *Programmer's Guide*, available in the electronic annex on the application CD-ROM.

The program

This next section takes you step by step through FRSTPRM.VBP. Each command is discussed and, where necessary, references point you to the corresponding entry in the *Programmer's Guide*.

1 Initializing variables

MyApp refers to the **Application** object, representing the label design software. This object is at the top of the hierarchy.

MyDoc refers to the **Document** object, representing the current label file.

MyVars refers to the **Variable collection**. This collection is a list object, and allows the document variables to be accessed.

Code:

```
Option Explicit
Dim MyApp As LabelManager2.Application
Dim MyDoc As LabelManager2.Document
Dim MyVars As LabelManager2.Variables
```

2 Creating objects

When the window is first opened, the objects are created and referenced (Set).

New LabelManager2.Application: creation of the **MyApp** object that represents the **Application**. This is the standard method for accessing and retrieving objects lower down in the hierarchy.

Code:

```
Private Sub Form_Load()
  Me.MousePointer = vbHourglass
  Set MyApp = New LabelManager2.Application
  MyApp.Visible = False
  Me.MousePointer = vbDefault
End Sub
```

3 Opening a document

Using the **Open** method on the **Documents** collection.

(The document name and access path are displayed in the text zone.) Calls the SelectFile function (see Chapter 10 - 114). Here, a condition test is applied to the document format. If the result is true, SelectFile opens the dialog box and applies the default format of the documents to be opened.

Once a document is selected, the **Documents** collection is called. When the document is opened, the **MyDoc** object is initialized and references the current document in the label design software.

Code:

```
Private Sub cmdOpen_Click()
 If SelectFile(”lab (*.lab)|*.lab”, 1) Then
  txtFilename = CommonDialog1.FileName
  Me.MousePointer = vbHourglass
  Set MyDoc = MyApp.Documents.Open(txtFile
   name)


  Me.MousePointer = vbDefault
 End If
End Sub
```

4 Saving the document

Using the **SaveAs** method on the **MyDoc** object.

Code:

```
Private Sub cmdSave_Click()
 If SelectFile(”lab (*.lab)|*.lab”, 2) Then
  txtFilename = CommonDialog1.FileName
  MyDoc.SaveAs txtFilename
 End If
End Sub
```

5 Opening a query

Using the **OpenQuery** method on the **Database** object. This method opens and executes a query in the current document.

Code:

```
Private Sub cmdOpenQuery_Click()
  If SelectFile(”csq (*.csq)|*.csq”, 1) Then
    txtQueryname = CommonDialog1.FileName
    Me.MousePointer = vbHourglass
    MyDoc.Database.OpenQuery txtQueryname
    Me.MousePointer = vbDefault
  End If
End Sub
```

6 Closing the query

Using the **Close** method on the **Database** object. This method closes the current query connection.

Code:

```
Private Sub cmdClosequery_Click()
  MyDoc.Database.Close
  txtQueryname.Text = ””
  lstVariables.Clear
End Sub
```

7 Preview of the current document

Calls the Preview procedure that opens the current document preview window (see Chapter 10 - 113 below).

Code:

```
Private Sub cmdPreview_Click()
   Preview
End Sub
```

8 Access to the variable collection

Displaying the current document variables.

Initializing the **MyVar** object that references the current document **Variables** collection, then browsing the collection in order to retrieve the variable names.

Code:

```
Private Sub cmdVariables_Click()
  Dim var As LabelManager2.Variable
  lstVariables.Clear
  Set MyVars = MyDoc.Variables
    For Each var In MyVars
      lstVariables.AddItem var.Name
    Next
End Sub
```

9 Accessing the form

Accessing the form in the label design software requires the use of the **Dialogs** collection (dialog box collection). The name of the form in this collection is **LppxFormDialog**. Activating the form uses the **Show** method.

Code:

```
Private Sub cmdFiller_Click()
  MyApp.Dialogs(lppxFormDialog).Show
End Sub
```

10 Adding printers

Accessing the label design software **Printers** dialog box requires the use of the **Dialogs** collection. The name of the dialog box in this collection is **LppxPrinterSelectDialog**. Activating this dialog box uses the **Show** method.

Code:

```
Private Sub cmdAddPrinter_Click()
   MyApp.Dialogs(lppxPrinterSelectDialog).Show
End Sub
```

11 Printing the document

Using the **PrintLabel** method to launch printing, and the **FormFeed** method to terminate printing on the **MyDoc** object.

Code:

```
Private Sub cmdPrint_Click()
   MyDoc.PrintLabel txtQuantity
   MyDoc.FormFeed
End Sub
```

12 Print merge

Using the **Merge** method on the **MyDoc** object. By default, merge launches printing of a single label series.

Code:

```
Private Sub cmdMerge_Click()
   Mydoc.Merge 1
End Sub
```

13 Access to the label design software: ”No visible” option

Setting the **MyApp** object **Visible** property to **False**. By default, the label design software is invisible (first option button).

Code:

```
Private Sub optNoControl_Click()
   MyApp.Visible = False
End Sub
```

14 Access to the label design software: ”Visible” option

Setting the **MyApp** object **Visible** property to **True**. The label design software becomes visible (second option button).

This procedure includes an example of OLE error management. Any **ActiveX** operation (accessing a property or calling a function) is susceptible to generating an error. The error is represented by a code that can be accessed using the **GetLastError** method, or by the associated **MyApp** object **ErrorMessage** method.

Code:

```
Private Sub optLppx_Click()
  MyApp.Visible = True
  ’sample of OLE error management :
  Dim Errornum As Long, ErrorMsg As String
  Errornum = MyApp.GetLastError
  ErrorMsg = MyApp.ErrorMessage(Errornum)
  If Errornum <> 0 Then
     MsgBox ErrorMsg, vbCritical, ”Error #”
        & Errornum


  End If
  ’sample end here
End Sub
```

15 Access to the first record in the database

Using the **MoveFirst** method on the **Database** object. This option accesses the first record in the database.

Code:

```
Private Sub CmdFirstRecord_Click(Index As Inte-
ger)
   MyDoc.Database.MoveFirst
End Sub
```

16 Access to the previous record in the database

Using the **MovePrevious** method on the **Database** object. This option accesses the previous record in the database.

Code:

```
Private Sub CmdPreviousRecord_Click(Index As Integer)
   MyDoc.Database.MovePrevious
End Sub
```

17 Access to the next record in the database

Using the **MoveNext** method on the **Database** object. This option accesses the next record in the database.

Code:

```
Private Sub CmdNextRecord_Click(Index As Integer)
   MyDoc.Database.MoveNext
End Sub
```

18 Access to the last record in the database

Using the **MoveLast** method on the **Database** method. This option accesses the last record in the database.

Code:

```
Private Sub CmdLastRecord_Click(Index As Integer)
   MyDoc.Database.MoveLast
End Sub
```

19 Preview of the current document

Calls the **Preview** procedure (see Chapter 10 - 113 below). Provides a preview of the document status allowing you to browse the database with the navigation bar.

Code:

```
Private Sub CmdPreview2_Click()
   Preview
End Sub
```

20 Preview procedure

Using the **CopyToClipboard** method on the **MyDoc** object. This method copies the document image onto the clipboard. The image can then be retrieved using the Visual Basic Clipboard object.

Code:

```
Sub Preview()
   Mydoc.CopyToClipboard
   Load frmPreview
   frmPreview.imgPreview.Picture=Clipboard.Get-
Data(vbCFMetafile)
   frmPreview.Show
End Sub
```

21 SelectFile procedure

This procedure filters the expected file format when the file is opened or closed, i.e. the lab format for a document or the csq format for a query.

Code:

```
Private Function SelectFile (ByVal filters As
Sring, ByVal ActionCode As Long) As Boolean
   On Error Resume Next
   With CommonDialog1
      .Filter=filters
      .FileName=Mid(filters,InStrRev(fil-
ters,"|")+1)
      .Action=ActionCode
      'Dialog box is displaying
      '...
      'Dialog box is closed
      If Err = 0 Then
         SelectFile = True
      End If
   End With
End Function
```

22 Exit the application

To shut the label design software down properly, you need to:

a. Close all documents, using the **CloseAll** method on the **Documents** collection.

b. Use the **Quit** method on the **MyApp** object (thus telling the label design software that processing is finished).

c. Ask Visual Basic to remove the label design software from the system memory.

Code:

```
Private Sub cmdQuit_Click()
  Unload Me
End Sub

Private Sub UnloadLppx()
  MyApp.Documents.CloseAll False
  MyApp.Quit
  Set MyApp = Nothing

End Sub
```

23 To exit the application using the title bar

The procedure is the same.

(This procedure is added if the user does not use the **Quit** button to close the application, as it is important the process is completed.)

Code:

```
Private Sub Form_Unload(Cancel As Integer)
  UnloadLppx
  Unload frmPreview
End Sub
```

⚠ It is vital that the above process for ending processing (quitting the application) is followed if the system is to be left in a stable state.

Memo

11

My variable displays a truncated result

If the result is truncated, check the settings in the **Output** tab of the dialog box for the variable in question. The **Maximum length** value is too low for the whole value of your variable to be displayed.

You need to create intermediate variables

Consider using the **Free** branch in the **Document Browser Data Sources** tab.

ODBC and reserved words

Warning: certain words are reserved for ODBC functions, and must not be used to name your data sources. Refer to the list of reserved words below:

ABSOLUTE
ADA
ADD
ALL
ALLOCATE
ALTER
AND
ANY
ARE
AS
ASC
ASSERTION
AT
AUTHORIZATION
AVG
BEGIN
BETWEEN
BIT
BIT_LENGTH
BY
CASCADE
CASCADED
CASE
CAST
CATALOG
CHAR
CHAR_LENGTH
CHARACTER
CHARACTER_LENGTH
CHECK
CLOSE
COALESCE
COBOL
COLLATE
COLLATION
COLUMN
COMMIT
CONNECT
CONNECTION
CONSTRAINT
CONSTRAINTS
CONTINUE
CONVERT
CORRESPONDING
COUNT
CREATE
CURRENT
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP
CURSOR
DATE
DAY
DEALLOCATE
DEC
DECIMAL
DECLARE
DEFERRABLE
DEFERRED
DELETE
DESC
DESCRIBE
DESCRIPTOR
DIAGNOSTICS
DICTIONARY
DISCONNECT
DISPLACEMENT
DISTINCT
DOMAIN
DOUBLE
DROP
ELSE
END
END–EXEC
ESCAPE
EXCEPT
EXCEPTION
EXEC
EXECUTE
EXISTS
EXTERNAL
EXTRACT
FALSE
FETCH
FIRST
FLOAT
FOR
FOREIGN
FORTRAN
FOUND
FROM
FULL
GET
GLOBAL
GO
GOTO
GRANT
GROUP
HAVING
HOUR
IDENTITY
IGNORE
IMMEDIATE
IN
INCLUDE
INDEX
INDICATOR
INITIALLY
INNER
INPUT
INSENSITIVE
INSERT
INTEGER
INTERSECT
INTERVAL
INTO
IS
ISOLATION
JOINKEY
LANGUAGE
LAST
LEFT
LEVEL
LIKE
LOCAL
LOWER

MATCH
MAX
MIN
MINUTE
MODULE
MONTH
MUMPS
NAMES
NATIONAL
NCHAR
NEXT
NONE
NOT
NULL
NULLIF
NUMERIC
OCTET_LENGTH
OF
OFF
ON
ONLY
OPEN
OPTION
OR
ORDER
OUTER
OUTPUT
OVERLAPS
PARTIAL
PASCAL
PLI
POSITION
PRECISION
PREPARE
PRESERVE
PRIMARY
PRIOR
PRIVILEGES
PROCEDURE
PUBLIC
RESTRICT
REVOKE
RIGHT
ROLLBACK
ROWS
SCHEMA
SCROLL
SECOND
SECTION
SELECT
SEQUENCE
SET
SIZE
SMALLINT
SOME
SQL
SQLCA
SQLCODE
SQLERROR
SQLSTATE
SQLWARNING
SUBSTRING
SUM
SYSTEM
TABLE
TEMPORARY
THEN
TIME
TIMESTAMP
TIMEZONE_HOUR
TIMEZONE_MINUTE
TO
TRANSACTION
TRANSLATE
TRANSLATION
TRUE
UNION
UNIQUE
UNKNOWN
UPDATE
UPPER
USAGE
USER
USING
VALUE
VALUES
VARCHAR
VARYING
VIEW
WHEN
WHENEVER
WHERE
WITH
WORK
YEAR

Forbidden characters and variable names

The characters forbidden for naming variables are: ”**{**” and ”**}**”. Moreover, if a formula includes a variable, the name of which contains any of the following characters: **&+-*/<>=^%,!\”**, the variable name must be enclosed within the **{ } characters.**

Formula0: Import0 & Import1
Formula1: Import0 & {toto%titi}

MDI interface

Your label design software interface is MDI, that means you can open a number of documents at the same time.

Note

The different documents open at the same time will all share the same interface settings (grid, language, and so on). However, the selected printer, merged database and the form are specific to each one.

Glossary

12

ActiveX **ActiveX** is an open integration platform providing a simple and effective solution for your integration needs.

Database A database allows you to store data. All data is organized into two-dimensional **tables**, called a **relationship**. Each row in the table is called a **record**. The purpose of a record is to manage an object, the properties of which are located in the different columns of the database table in the form of **fields**.

Fields Fields are contained within a database table and allow object properties to be managed.

Record The purpose of a record is to manage an object, the properties of which are located in the different columns in the database table in the form of **fields**.

Function
Functions are ready-to-use formulas, and can be very complex. A function can be integrated into a formula.

Formula
A formula is an equation that analyses data using a given expression. Formulas carry out operations such as addition, multiplication, comparison of values and date calculations. They can also incorporate variables, functions, operators, and so on.

Join
Joins allows a number of tables in the same database to be linked. There are three types of join:
- internal join;
- right join;
- left join.

Variable object
A variable object is a variable that takes on a physical appearance, either text, a barcode or an image.

ODBC
Open Data Base Connectivity is a Microsoft product that provides a system of database access. ODBC makes connecting your label design software to a certain number of databases very easy.

ODBC data source
The data source contains both the data and the associated database engine.

Variable
A variable is a non-physical object, the value of which can vary. A variable can belong to different **data sources**:
- Database;
- Table lookup;
- Formula;
- Counter;
- Date;
- Form.

Database variable
Database variable refers to a variable extracted from the database merged with the current label.

Counter variable

A **Counter** is a variable, the value of which advances from one label to another during printing, according to an initial value and an increment. Often used for numbering a series of labels in a print run, the counter can be calculated by the computer or the printer.
There are six calculation bases:
- **Binary**: base 2 calculation;
- **Octal**: base 8 calculation;
- **Decimal**: base 10 calculation;
- **Hexadecimal**: base 16 calculation;
- **Alphabetical**: base 26 calculation (A to Z);
- **Alphanumeric**: base 36 calculation (0 to 9 and A to Z)
- **Custom**: calculation base chosen by the user.

Date variable

The **Date** variable allows the date and time to be automatically inserted into your document. The format can be completely modified and the value is updated by the system.

Variable Form

The content of a **Form** variable is the result of keyboard input.

Formula variable

The content of a **Formula** variable is the result of a calculation.

Table lookup variable

Database lookup variables refer to variables extracted from a database other than the one merged with the current label.

Control variables

These variables are part of your application and are used to trigger special functions when printing is launched.

@LABNAME This variable refers to the name of the current document. The field that is given this name determines the name of the document to be loaded during an import operation.

@SERIALQTY The field that is given this name determines the number of labels to be printed. It must be numerical.

@LABELCPY The field that is given this name determines the number of identical labels to be printed between counter increments (if the document contains one or more counters). It must be numerical.

@PAGECPY The field that is given this name determines the number of copies of each page to be printed. It must be numerical.

@INTERCUT The field that is given this name determines the number of labels to be printed between each cut. It must be numerical.

Index

13

www.zebra.com

Zebra Technologies Corporation

International Headquarters
333 Corporate Woods Parkway
Vernon Hills, IL 60061 U.S.A.
Phone: +1 847.634.6700
Fax: +1 847.913.8766

Zebra Technologies Europe Limited

European Headquarters
Zebra House
The Valley Centre, Gordon Road
High Wycombe
Buckinghamshire HP13 6EQ, UK
Phone: +44.1494.472872
Fax: +44.1494.450103

Zebra Technologies Singapore

Regional Sales Office-Asia/Pacific
1 Sims Lane, #06-11
Singapore 387355
Phone: +65.842.0322
Fax: +65.842.0366

Other Locations

United States: California, Florida,
Rhode Island, Wisconsin
Europe: France, Germany, Italy
Asia: Hong Kong, Japan, Seoul

Zebra Bar Code Products