TEKLYNX®

CODESOFT® V. 9

TUTORIAL

TEKLYNX

About this manual

Typographical conventions

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

- Terms taken from the interface itself, such as commands, appear in **bold**.
- Keys appear in small caps. For example: ''Press the SHIFT key''.
- Numbered lists indicates that there is a procedure to follow.
- 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 appear in bold. Thus, ''Go to **File > Open**'' means choose the **File** menu then the **Open** command.

About your product

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

For the complete list of specific features available in your software, refer to the specification sheet provided with the product.

TABLE OF CONTENTS

Connecting to database

A few reminders

In this chapter we are going to link a label (the container) with a database (the content). To do this, we will use ODBC (Open DataBase Connectivity) or OLE DB connections.

Databases allow you to store data. All data is organized into two-dimensional tables in what is 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 within a given database, we use joins. A concrete example later in this chapter demonstrates how joins are created.

ODBC

ODBC data sources make it possible to access data belonging to a wide range of database management systems. ODBC makes it easy to link an application such as your label design software with a certain number of databases. The software comes with several ODBC drivers.

These enable you to access the most common types of databases.

Some of the drivers are listed below:

- Microsoft Access Driver (*.mdb)
- Microsoft Excel Driver (*.xls)
- Microsoft FoxPro Driver (*.dbf), etc.

OLE DB

OLE DB is a set of interfaces that gives access to all data, regardless of type, format or location. It provides components such as access interfaces, query drivers, and so on. These components are called "providers".

Practical Workshop 1

Installing an ODBC data source and importing data

The example below describes a connection process when a database is not connected to your software.

Installing an ODBC data source

The process described below uses the direct creation mode. If you want, you can use the Wizard by selecting Wizard in the context menu.

Connecting to the TKTraining.mdb database

1. In Codesoft, choose **Tools > Administrator ODBC**.

2. Click on the **System Data Source** tab (DSN) then click **Add**.

Note: You can define data sources with system Data Sources Names (DSNs). These data sources are unique to a specific computer but not to a specific user. Any user with the necessary rights can access a system DSN.

3. Select Microsoft Access Driver then click Finish.

4. Enter ''TK Training CS Level 2'' in the Data source name box.

5. Click Select and select the database TKTraining.mdb, located in C:\CS TRAINING\LEVEL 2\DATA.

6. Click the **Options** button. Check the Read Only option. This option makes it possible to open the database at the same time as Codesoft without causing any read/write problems.

7. Click **OK** in the **ODBC Microsoft Excel Setup** dialog box.

Importing data

When the database is connected to your software, you have to connect it to your document.

1. Open the label PRODUCT_WS3.

2. Choose **Data source > Database > Create/Modify query**.

3. Select TK Training CS Level 2 in the Select data source list.

4. Select "Fruits" in the Select table list.
   The database fields appear in the Select fields list.

5. Select the "ProdName", "Origin", "Weight", and "Reference" fields.

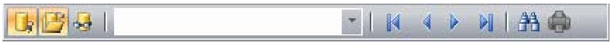
6. Click the button. It Allows the selected records to be sorted into alphabetical or numeric order, Ascending, or Descending.

7. Select "Reference" as the **Sort Key** and "Ascending" as the **Sort Order**.

8. Save the query as C:\CS TRAININGLLEVEL 2\CSQ\PRODUCT_WS4_ODBC.CSQ.

9. Click **OK**.
   The variables are created automatically and are listed in the Database branch in the Data sources view.

To view or print the different values that your object can take, use the navigation bar. You can also print from the **Query result** window.



Creating variable objects

1. Select the created variables listed in the Database branch in the Data sources view, then drag and drop it into the workspace.

2. Select **Text** in the context menu.

Command: **Data source > Table lookup > Properties**.

1. Select the data source from the **Select data source** list.

> Note: To create a new data source, click the New data source button. This gives you the option of using the Wizard or choosing between an ODBC or OLEDB data source.

2. By default, **Standard** creation mode is activated. However to carry out a table lookup, you can use **Advanced** creation mode: SQL.

Standard creation mode

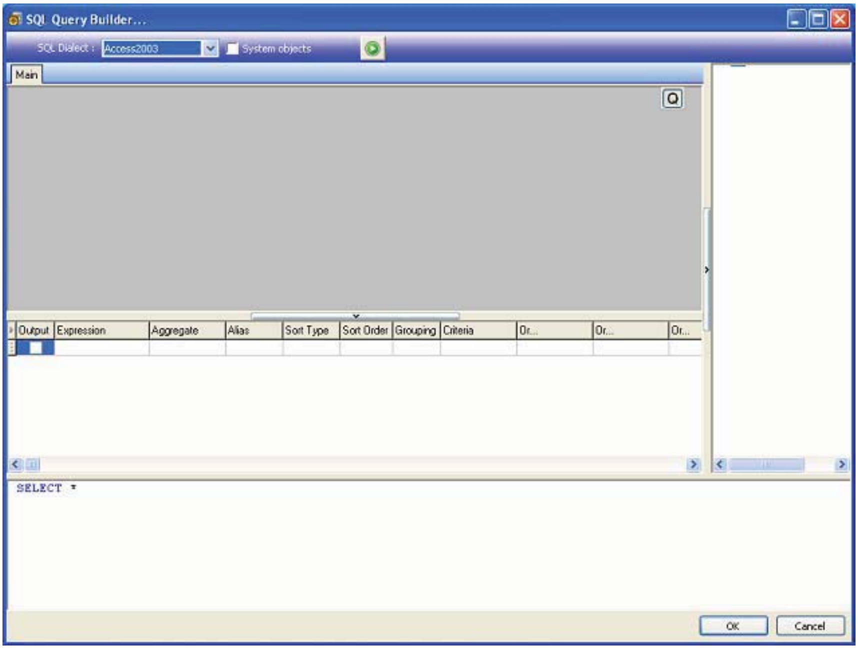
3. In the **Select table** list, select a table where the search is to be carried out.
4. In the **Select result field** list, select the field whose value is to be transferred into your variable.
5. Click to add a row.
6. Select the field in the external table on which the search is to be carried out.
7. Select the current document variable containing the search value.
8. Click the **Test** button to display the result.

SQL advanced creation mode

3. Click **SQL** format creation mode.
4. Enter your query in SQL format.

   - or -

   Click **SQL Query Builder** to access Query Builder. This provides an easy-to-use interface for building SQL database queries. You can create new requests or represent existing requests graphically in your applications.



```
SELECT *
```

5. Click the **Test** button to display the result in the **Query** dialog box.

Query Builder

Active Query Builder is a visual query builder component that allows you to build complex SQL queries via an intuitive visual query building interface.

To work with Active Query Builder, you need basic knowledge of SQL concepts. Active Query Builder will help you to write correct SQL code hiding technical details, but only understanding of the SQL principles will make it possible to achieve desired results.

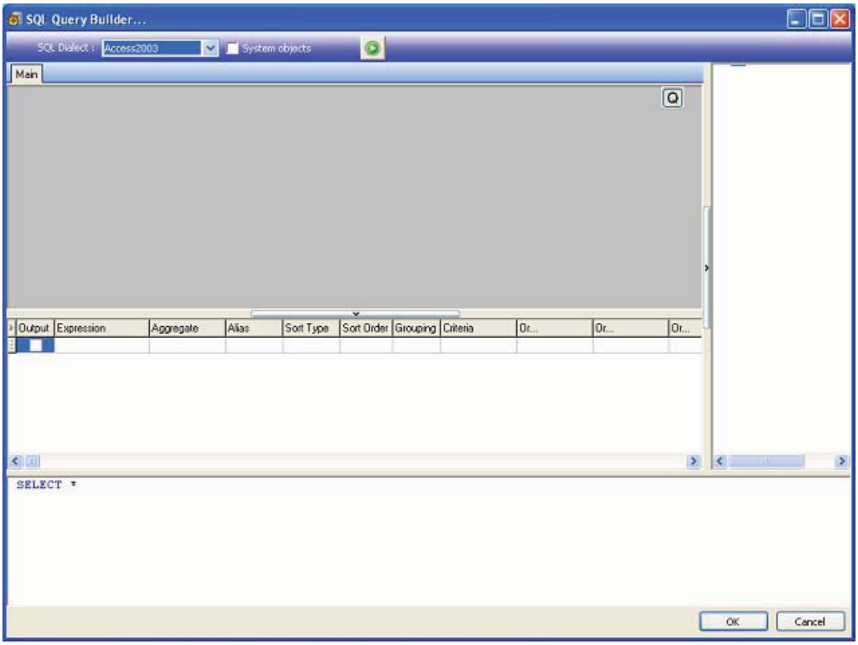
Getting started

This is how Active Query Builder looks when started.

The main window can be divided into the following parts:

- The **Query Building Area** is the main area where the visual representation of query will be displayed. This area allows you to define source database objects and derived tables, define links between them and configure properties of tables and links.
- The **Columns Pane** is located below the query building area. It is used to perform all necessary operations with query output columns and expressions. Here you can define field aliases, sorting and grouping, and define criteria.

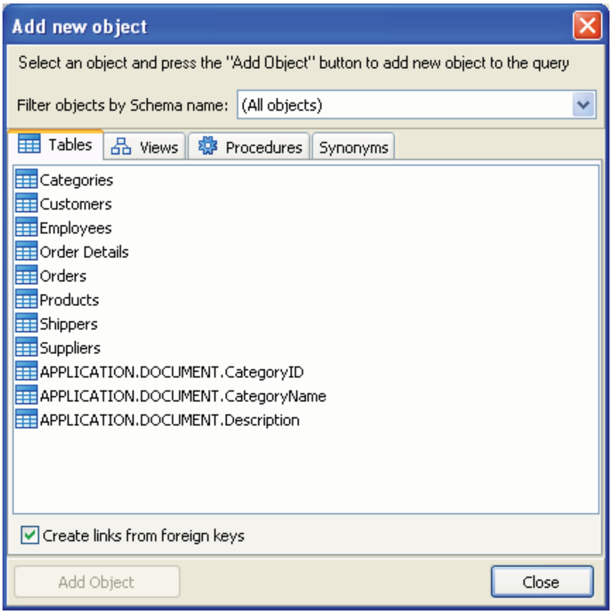
- The **Query Tree Pane** is located at the left. Here you may browse your query and quickly locate any part of it.
- The page control above the query building area will allow you to switch between the main query and sub-queries.
- The small area in the corner of the query building area with the "Q" letter is the union sub-query handling control. Here you may add new union sub-queries and perform all necessary operations with them.



```
SELECT *
```

Adding an object to the query

To add an object to the query right click the query building area and select the Add Object item in the drop-down menu.


Add new object
Select an object and press the "Add Object" button to add new object to the query
Filter objects by Schema name: (All objects)
Tables
Views
Procedures
Synonyms
Categories
Customers
Employees
Order Details
Orders
Products
Shippers
Suppliers
APPLICATION.DOCUMENT.CategoryID
APPLICATION.DOCUMENT.CategoryName
APPLICATION.DOCUMENT.Description
Create links from foreign keys
Add Object
Close

The **Add New Object** window allows you to add as many objects as you wish at once. The objects are grouped according to their type by four tabs: **Tables, Views, Procedures** (Functions) and **Synonyms**. You may select one or several objects by holding the **Ctrl** key and then press the **Add Object** button to add these objects to the query. You may repeat this operation several times. After you finish adding objects, press the **Close** button to hide this window.

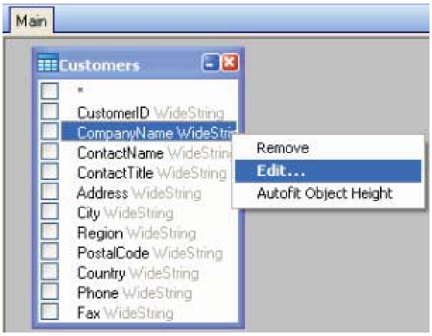
To remove an object from the query, select it and press the **Del** key or simply click the **Close** button in the object header.

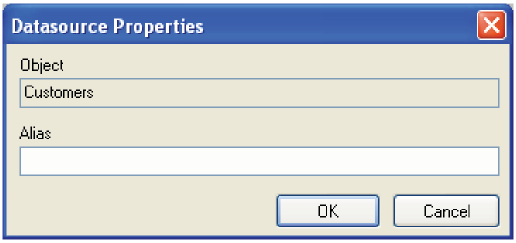
For those servers that have schemas or allow selection of objects from different databases, you may filter objects by database or schema name by selecting the necessary schema or database from the combo box at the top of the window.

Active Query Builder can establish links between tables based on the information about foreign keys in the database. This ability is turned on by default. To turn it off, uncheck the **Create links from foreign keys** checkbox.

Editing object properties

You may change the properties of each object added to the query by right clicking the object and selecting the **Edit...** item from the drop-down menu or simply by double-clicking the object header.




Datasource Properties

Object

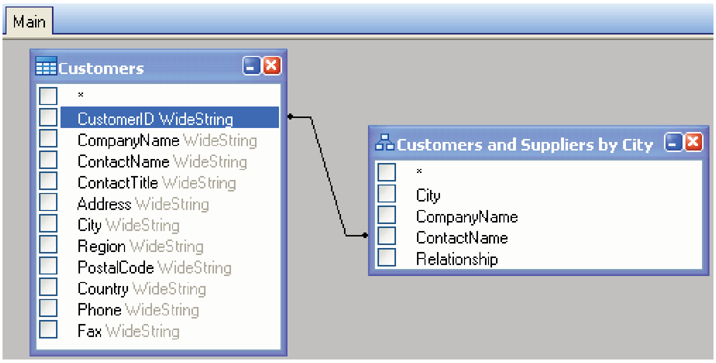
Customers

Alias

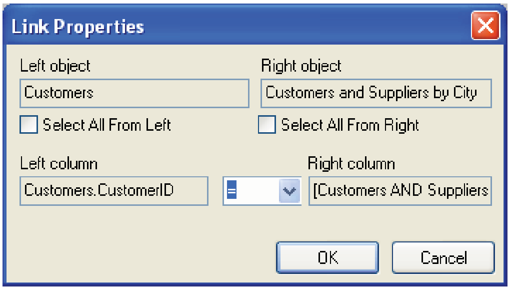
The **Datasource Properties** dialog may vary from server to server, but at least the Alias property is the same for all database servers.

Joining tables

To create a link between two objects (i.e. join them) you should select the field by which you want to link the object with another and drag it to the corresponding field of the other object. After you finish dragging, a line connecting the linked fields will appear.



The join type created by default is INNER JOIN, i.e. only matching records of both tables will be included in the resulting dataset. To define other types of joins you should right click the link and select the **Edit...** item in the drop down menu or simply double-click it to open the **Link Properties** dialog. This dialog allows you to define join type and other link properties.



To remove a link between objects, right-click the link line and select the **Remove** item in the drop-down menu.

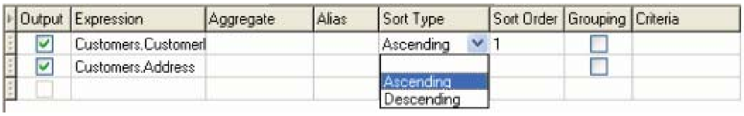
Sorting output fields

To enable sorting of output query fields you should use the **Sort Type** and **Sort Order** columns of the **Columns Pane**.

The **Sort Type** column allows you to specify the way the fields will be sorted - in the **Ascending** or **Descending** order.

The **Sort Order** column allows you to setup the order in which fields will be sorted, if more than one field will be sorted.

To disable sorting by some field you should clear the **Sort Type** column for this field.



Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria
✓	Customers.CustomerI			Ascending	1	☐	
✓	Customers.Address					☐	
☐							

Defining criteria

To define criteria for the expression listed in the **Columns Pane** you must use the **Criteria** column.

There you should write the criterion omitting the expression itself. To get the following criterion in your query

WHERE (field >= 10) **AND** (field <= 20)

you should write

>= 10 **AND** <= 20

in the **Criteria** column.

You may specify several criteria for a single expression using the **Or...** columns. These criteria will be concatenated in the query using the **OR** operator.

The Query result grid

To access the Query result grid, click the button in the Defining a query dialog box, in the Merge database browser toolbar or via the Data source > Databases menu.

This grid allows the result of a query to be displayed, or to search for a particular term and all its occurrences and print the corresponding labels.

The **Query result** grid contains:

- **Search functions**

  Search field, allowing the field in which the search will be carried out in to be entered.

  Data to search for, allowing the value to be searched for to be entered.

  Search for the value anywhere in the field or at the start of the field .

- **Navigation functions for browsing query result records**

  First record

  Previous record

  Next record

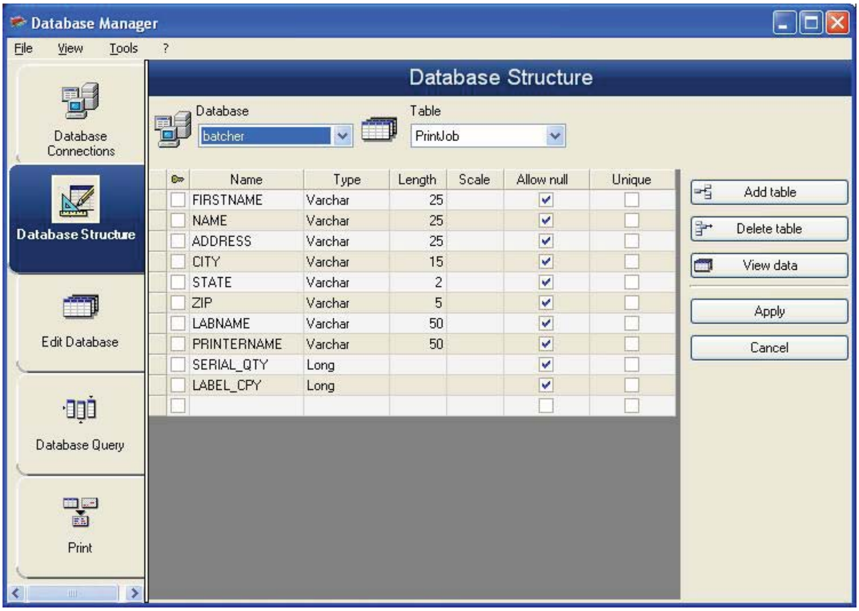
  Last record

The results grid
Displays the occurrences of the query result.

Requery
Requery the request and update the grid.

Database Manager

Database Structure Window



Database: batcher

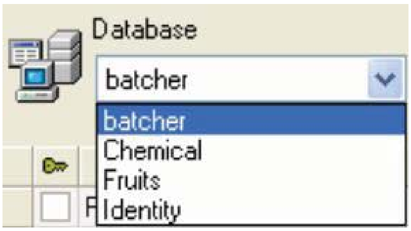
Table: PrintJob

🔑	Name	Type	Length	Scale	Allow null	Unique
☐	FIRSTNAME	Varchar	25		✓	☐
☐	NAME	Varchar	25		✓	☐
☐	ADDRESS	Varchar	25		✓	☐
☐	CITY	Varchar	15		✓	☐
☐	STATE	Varchar	2		✓	☐
☐	ZIP	Varchar	5		✓	☐
☐	LABNAME	Varchar	50		✓	☐
☐	PRINTERNAME	Varchar	50		✓	☐
☐	SERIAL_QTY	Long			✓	☐
☐	LABEL_CPY	Long			✓	☐
☐					☐	☐

The Database Structure window is used to manage the structure of the database file: to add, modify or delete tables/ fields etc.

Choose a database from the list of connections

1. Click on the **Database** drop-down list.
2. Click on the data required.

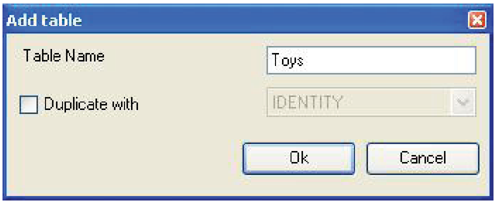


Choose a table in a database

1. Click on the **Table** drop-down list.
2. Click on the data required.

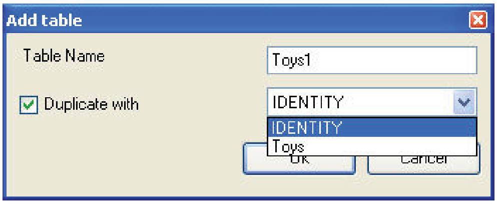
Add a table to the active database

1. Click **Add table**.
2. Enter the name of the new table.
3. Click **OK**.



You can also copy the structure of the table from a table that already exists in the selected database. To do so:

1. Tick the box next to **Duplicate with**.
2. Click on the drop-down list.
3. Click on the data required.
4. Click **OK**.



Delete a table in the active database

1. Click on the **Table** drop-down list.
2. Click on the data required.
3. Click **Delete** table.

View/hide active table’s data

1. Click View data.

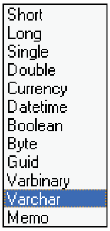
Define a key field

1. Tick the box next to the required field.

2. Click **Apply**.

Define a field's type of content

1. Click on the required field in the **Type** column.
2. Click the drop-down list button.
3. Click on the data required.



4. Click **Apply**.

Define a field's maximum size

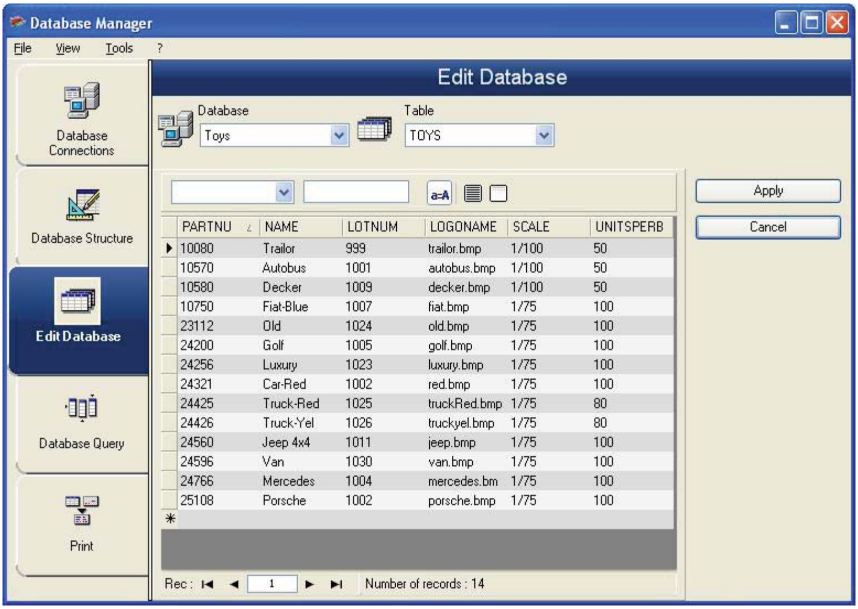
1. Click on the required field in the **Length** column.
2. Enter the value required.

3. Click **Apply**.

Allow an empty field

1. Tick the Allow **Null box** for the required field.
2. Click **Apply**.

Edit Database window



PARTNU	NAME	LOTNUM	LOGONAME	SCALE	UNITSPERB
10080	Trailor	999	trailor.bmp	1/100	50
10570	Autobus	1001	autobus.bmp	1/100	50
10580	Decker	1009	decker.bmp	1/100	50
10750	Fiat-Blue	1007	fiat.bmp	1/75	100
23112	Old	1024	old.bmp	1/75	100
24200	Golf	1005	golf.bmp	1/75	100
24256	Luxury	1023	luxury.bmp	1/75	100
24321	Car-Red	1002	red.bmp	1/75	100
24425	Truck-Red	1025	truckRed.bmp	1/75	80
24426	Truck-Yel	1026	truckyel.bmp	1/75	80
24560	Jeep 4x4	1011	jeep.bmp	1/75	100
24596	Van	1030	van.bmp	1/75	100
24766	Mercedes	1004	mercedes.bm	1/75	100
25108	Porsche	1002	porsche.bmp	1/75	100

Rec : 1 Number of records : 14

The Edit Database window is used to manage the contents of the database file: to add, modify or delete data.

These actions depend on the type of database. Thus, Excel file records cannot be modified.

Select records according to their content

Use the content of a field to find a record.

1. Click the drop-down list button.
2. Click on the data required.
3. Click the data input field.
4. Enter the value required in the data input field.

Select all identical records.

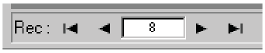
At least one record must have been found.

1. Click the drop-down list button.
2. Click on the data required.
3. Click the data input field.
4. Enter the data required in the data input field.
5. Click on the Select all button (▤)

Select an identical record.

At least one record must have been found. There must be several identical contents in the search field.

To select a record, use the search tool: click on 1 (First), 2 (Previous), 3 (Next) or 4 (Following).



Create a new record

1. Click on a field in the row marked with an asterisk.
2. Enter the values required in the corresponding fields.
3. Click **Apply**.

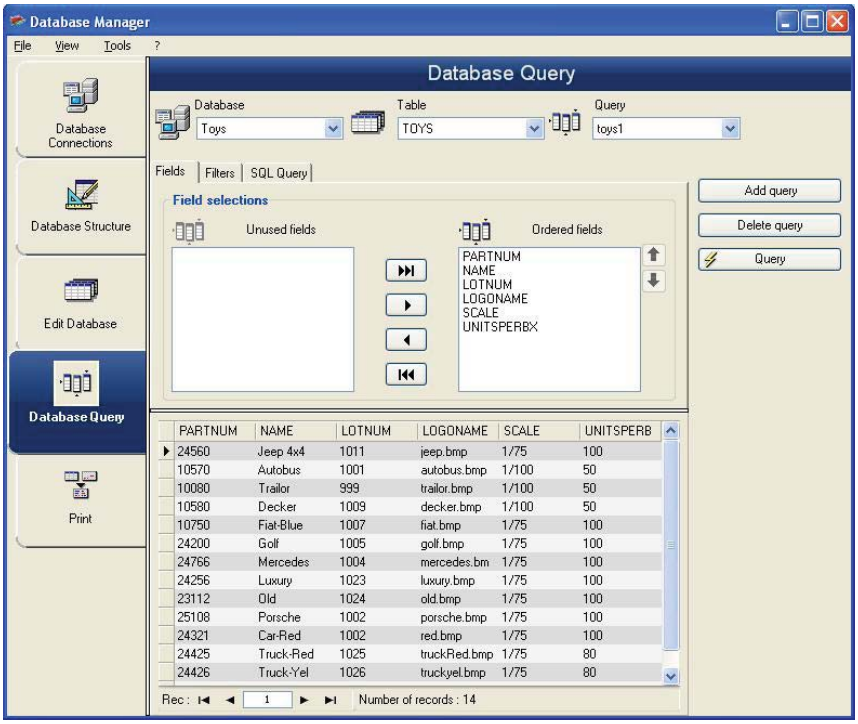
Modify a record

1. Click on the data you want to modifiy.
2. Enter the data required.
3. Click **Apply**.

Delete a record

1. Click the database cursor for the required field.
2. Right click the database cursor for the required field.
3. Click **Delete Record** in the context menu.

Database Query window



Ordered fields: PARTNUM, NAME, LOTNUM, LOGONAME, SCALE, UNITSPERBX

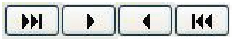
PARTNUM	NAME	LOTNUM	LOGONAME	SCALE	UNITSPERB
24560	Jeep 4x4	1011	jeep.bmp	1/75	100
10570	Autobus	1001	autobus.bmp	1/100	50
10080	Trailor	999	trailor.bmp	1/100	50
10580	Decker	1009	decker.bmp	1/100	50
10750	Fiat-Blue	1007	fiat.bmp	1/75	100
24200	Golf	1005	golf.bmp	1/75	100
24766	Mercedes	1004	mercedes.bm	1/75	100
24256	Luxury	1023	luxury.bmp	1/75	100
23112	Old	1024	old.bmp	1/75	100
25108	Porsche	1002	porsche.bmp	1/75	100
24321	Car-Red	1002	red.bmp	1/75	100
24425	Truck-Red	1025	truckRed.bmp	1/75	80
24426	Truck-Yel	1026	truckyel.bmp	1/75	80

The Database Query window is used to create and apply various filters.

Add a query

1. Click **Add query** in the **Fields** tab.
2. Enter a name for the query.
3. Click **OK**.

Select/deselect one or more fields

1. Use the navigation tool .
2. Click **Query** to refresh the database preview.

Modify the order of fields selected

1. Click on the required field in the **Ordered fields** window.
2. Click the **Up** or **Down** arrow to reach the data required.
3. Click **Query** to refresh the database preview.

Create a filter using predefined data

1. Select **Filters** tab.
2. Click on the Add row button ().
3. In the **Field** field, select in the drop-down list the data required.
4. In the **Operator** field, select in the drop-down list the operator required.
5. In the **Value** field, enter the value required.
6. Click **Query** to view the result.

Apply a logical operator to several filters

1. Click on the Add row button ().
2. In the **Logical** field select in the drop-down list the data required (AND or OR).
3. Create a filter (as described above).
4. Click **Query** to view the result.

Remove a filter

Note: At least one filter must exist.

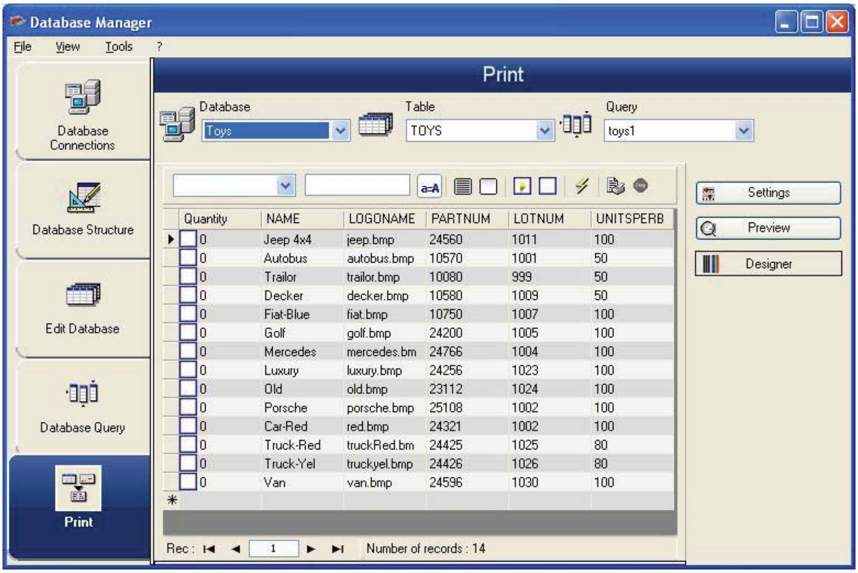
1. Click on the database cursor for the required field.
2. Click Delete row button ().

Modify a filter in SQL

Note: At least one filter must exist.

1. Select **SQL Query** tab.
2. Check **Modify the query in SQL language** to activate the SQL Query and make manual changes.
3. Click **Query** to view the result.

The Print window



Database: Toys
Table: TOYS
Query: toys1

Quantity	NAME	LOGONAME	PARTNUM	LOTNUM	UNITSPERB
0	Jeep 4x4	jeep.bmp	24560	1011	100
0	Autobus	autobus.bmp	10570	1001	50
0	Trailor	trailor.bmp	10080	999	50
0	Decker	decker.bmp	10580	1009	50
0	Fiat-Blue	fiat.bmp	10750	1007	100
0	Golf	golf.bmp	24200	1005	100
0	Mercedes	mercedes.bm	24766	1004	100
0	Luxury	luxury.bmp	24256	1023	100
0	Old	old.bmp	23112	1024	100
0	Porsche	porsche.bmp	25108	1002	100
0	Car-Red	red.bmp	24321	1002	100
0	Truck-Red	truckRed.bm	24425	1025	80
0	Truck-Yel	truckyel.bmp	24426	1026	80
0	Van	van.bmp	24596	1030	100

Rec : 1 Number of records : 14

The Print window is used to select files for printing, to assign printers and to define various parameters before printing is launched.

Select a document to be printed

1. Select a document from a file
2. Check File box in the Label name group.

   - or -

1. Click on the **Label Creation Wizard** button ( ).
2. Follow the wizard's instructions.

Note: Creating a label in relation to the database allows you to define exactly which elements are required to position each database field.

Select an existing label template

1. Click on the Open an existing document button ( ).
2. Select a .lab file.
3. Click **OK**.

Note: The ''Field'' radio buttons in the ''Label name'' and ''Printer name'' groups of options allow you to choose the label or printer required, when the latter are defined in one of the fields of the active database.

Select a document from a field

If your Database contains the name of the label to be used for the printing job in one of the fields, you can define this field as being the place where Database Manager is going to pick up the .lab file.

Ref	Designation	Qt	Code	Labname
6574	Ref1	1	9876546321	Label1.lab
6354	Ref2	2	1236478855	Label2.lab
6987	Ref3	3	6987456321	Label1.lab
3684	Ref4	4	3698745632	Label3.lab

In this example, the field 'Labname' can be used as Labname field.

1. Check Field box in the Label name group.
2. Select the field requiered.

Select a printer

Click on the **Add** or **Remove a printer** button ().

Formulas

The Formula data source

Command: Data source > Formula > Add

The **Formula** data source contains a list of data sources you have created. These data sources are populated by combinations of operators, constants, data sources, control variables, formulas, and functions. Data can be numeric or alphanumeric.

In order to carry out a calculation within a document, you must first create a Formula data source.

This data source has a specific dialog box allowing you to define the required function(s) for a given formula.

About functions

Functions are predefined formulas that carry out calculations by using particular values called arguments, in a certain order, called syntax.

Functions are used to return a numeric, character string, or logical value - the result of a calculation or an operation.

There are six groups of functions in the formula definition:

- Check character calculation functions
- Conversion functions
- Date and time functions
- Logical functions
- Mathematical functions
- Text/Character string functions

Operators

Operators are mathematical symbols indicating an operation to be carried out. There are different types of operators: arithmetic, comparative, concatenation and logical.

Arithmetic operators

Operator	Used for
*	Multiply two numbers.
+	Add two numbers together.
-	Subtract one number from another, or assign a negative value to an operand.
/	Divide one number by another.
^	Raise a number to the power of the exponent.
%	Modulo.

Comparative operators

Operator	Meaning
<	Less than.
<=	Less than or equal to.
>	Greater than.
>=	Greater than or equal to.
=	Equal to.
<>	Different from.

Concatenation operator

Used to combine two strings.

Operator	Meaning
&	Concatenation of two strings.

Logical operator

(see also logical functions)

Operator	Meaning
!	Not logical.

Mathematical functions

int (*value*): Returns the largest integer less than or equal to the *value* argument.

Examples:
int (-5.863) = -6
int (5.863) = 5

mod (*val_1*, *val_2*): Returns the remainder of the division of the *val_1* argument by the *val_2* argument. The result has the same sign as the divisor.

Examples:
mod (7,2) = 1
mod (-7,2) = -1
mod (7,-2) = 1
mod (-7,-2) = -1

quotient (*val_1*, *val_2*): Returns the integer result of the division of the *val_1* argument by the *val_2* argument.

round (*val_1*, *val_2*): Returns the argument *val_1* rounded to the number of figures indicated by *val_2*.

- If *val_2* is greater than 0, *val_1* is rounded to the number of decimals indicated.
- If *val_2* is equal to 0, *val_1* is rounded to the closest integer.

- If *val_2* is less than 0, *val_1* is rounded off to the left of the decimal point.

  Examples:
  round (4.25,1) = 4.3
  round (1.449, 1) = 1.4
  round (42.6,-1) = 40

trunc (*value*): Returns the integer part of the *value* argument.

base10tobaseX(*string_1,string_2*): Converts *string_2* from base 10 to base *string_1*

  Examples
  If the field named Base 16 contains the string "0123456789ABCDEF"
  BASE10TOBASEX(Base16, 12) produces C
  BASE10TOBASEX(Base16,10) produces A
  BASE10TOBASEX("012345","9") produces 13

Note: This formula cannot accept negative decimal numbers for ''string_2'' parameter.

baseXtobase10(*string_1,string_2*): Converts *string_2* from base *string_1* to base 10.

  Examples
  If the field named Base 16 contains the string "0123456789ABCDEF"
  BASEXTOBASE10(Base16, "E") produces 14
  BASEXTOBASE10(Base16,10) produces A
  BASEXTOBASE10("012345","9") produces 13

hex(*val_1,val_2*): Converts the *val_1* decimal number to hexadecimal format with a total value of *val_2*

Note: This formula cannot accept negative decimal numbers for ''val_1'' parameter.

Logical functions

Logical functions allow you to check whether one or more conditions have been fulfilled.

Note: TRUE equals 1 and FALSE equals 0.

and (*expr_1*, *expr_2*): Returns TRUE if both arguments are true, FALSE if at least one is false. The arguments must be calculated from logical values.

  Example:
  and(exact("string","string"),exact("string","string")) = 0
  and(exact("string","string"),exact("string","string")) = 1

exact (*string_1*, *string_2*): Returns TRUE if the two strings are identical, FALSE if not. This function is case-sensitive.

Example:
exact("software","software") = 1
exact("software","software") = 0

if (*expr*, *Val_if_true*, *Val_if_false*): Returns the *Val_if_true* value if *Expr* is true and the *Val_if_false* argument if *Expr* is false.

Example:
if(exact("string", "string"), "true", "false") = false
if(exact("string", "string"), "true", "false") = true

not (*logical*): Gives the opposite of the *logical* argument.

Example:
not(exact("string", "string")) = 1
not(exact("string", "string")) = 0
not(False) = 1 or not(0) = 1
not(True) = 0 or not(1) = 0
not(1+1=2) = 0

or (*expr_1*, *expr_2*): Returns TRUE if one of the two arguments is true and FALSE if both arguments are false. The arguments must be calculated from logical values.

Example:
or(exact("string", "string"),exact("string", "string")) = 0
or(exact("string", "string"),exact("string", "string")) = 1
or(true,true) = 1 or or(1,1) = 1
or(true,false) = 1 or or(1,0) = 1
or(false,false)= 0 or or(0,0) = 0

Text functions

A character string can be assimilated into a table if each box contains a character. It is defined by its length (total number of characters in the string, including spaces). The position of a character in the string corresponds to its place in the table. For example, the first character is at position one.

Example: Position 3 corresponds to the third character in the string.

cyclebasex *( ):* Allows counting to take place in any kind of database counting system. The numbering system must be defined within the linked expression. The start value, the value of each increment and the number of copies must also be specified for each number. All these values can be linked to other fields in the label, but the field names must not be enclosed in quotation marks.

Example:
If a field named Base 16 contains the character string 0123456789ABCDEF, then:
cyclebasex(base16, "8", 1 ,1) = 8,9,A,B,C&hellip;
cyclebasex(base16, "F", -1,1) = F,E,D,C,B,A 9,8,7&hellip;

cyclebasex(base16, "B0 ", 1,1) = B0, B1, B2&hellip;
cyclebasex("012345", "4",1,2) = 4,4,5,5,10,10,11,11&hellip;

cyclechar *( ):* Creates a user-defined set of characters for a complete cycle.

Example:
cyclechar("A", "C") = A B C A B C A B C &hellip;
cyclechar("A", "C", 1,2) = A A B B C C A A B B &hellip;

cyclenumber *( ):* Allows you to set your own sequence of numbers, instead of using the normal sequence of numbers or letters (0,1,2&hellip; or A,B,C&hellip;).

Example:
cyclenumber(1,3) will produce labels in the following sequence: 1 2 3 1 2 3 1 2 3...
cyclenumber(1,3,1,2) will produce labels in the following sequence: 1 1 2 2 3 3 2 2 1 1...

cyclestring *( ):* Allows you to create a group of words or characters using a complete cycle as an increment field. The whole string must be enclosed in quotation marks (" ") and each word or group of characters must be separated from the others by a semicolon (;).

Example:
cyclestring("Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun") = Mon Tue Wed Thu Fri Sat Sun
The following example is for labels that use all letters of the alphabet except for O and I.
cyclestring("A;B;C;D;E;F;G;H;J;K;L;M;N;P;Q;R;ST;U;V;W;X;Y;Z")

exact (*string_1*, *string_2*): Returns TRUE if the two strings are identical, FALSE if not.

Example:
exact("software","software") = 1
exact("sftware","software") = 0

extract («string», «sep», «pos»): Returns the substring from the character string «string» at the specified position «pos» that contains data separated by string «sep».

Example:
Extract("1;2;3;4", ";", 3) = 3

find (*string*, *key*, *start*): Returns the position of the first occurrence of the *key* argument in the *string* argument. The search in the *string* argument starts from the position returned by the *start* argument (*start* >= 1). The function resets to zero if no occurrence of the *key* argument is found. The function distinguishes between upper and lower case letters.

Example:
find("Peter McPeepert","P",1) = 1
find("Peter McPeepert","p",1) = 12

left (*string*, *num_char*): Returns the character string extracted from the *string* argument. This string starts at position one of the *string* argument and has a length equal to the *num_char* argument.

> Example:
> left("Peter McPeepert",1) = P
> left("Peter McPeepert ",5) = Peter

len (*string*): Gives the length of the *string* argument. Spaces are counted as characters.

> Example:
> len("Paris, New York") = 15
> len("") = 0
> len(" ") = 1

lower (*string*): Converts all upper case letters in a text string into lower case letters.

> Example:
> lower("Paris, New York") = paris, new york

mid (*string*, *start*, *num_char*): Returns the character string extracted from the *string* argument. This string starts at the position corresponding to the value of the *start* argument (*start* >=1) and has a length equal to the *num_char* argument.

> Example:
> mid("Paris, New York",8,8) = New York

pad *( ):* Adds characters to the left of the field to assign a predefined length to the whole input. Any character can be selected as a padding character.

> Example:
> If a field named GREETING displays a value HELLO, then:
> pad(GREETING,8,0) = 000HELLO
> pad(5,3,0) = 005
> pad(''Nine'',6,"a") = aaNine

replace (*string*, *start*, *num_char*, *new_string*): Returns the converted *string* argument. A number (equal to the *num_char* argument) of characters from the position defined in the *start* argument has been replaced by the *new_string* argument.

> Example:
> replace("Paris, New York",8,8,"Singapore") = Paris, Singapore

replaceString («string», «old_string», «new_string»): Replaces all occurrences of a specified «old_string» in character string «string» with another specified «new_string».

> Example:
> ReplaceString("abc12def12", "12", "") = abcdef

rept (*string*, *num_char*): Returns the character string where the *string* argument is repeated the number of times in the *num_char* argument.

> Example:
> rept("Ah Paris! ",2) = Ah Paris! Ah Paris!

right (*string*, *num_char*): Gives the character string composed of the last characters of the *string* and has a length equal to the *num_char* argument.

> Example:
> right("Purchase order",5) = order

search (*string*, *key*, *start*): Gives the position of the first occurrence of the *key* argument in the *string* argument. The search starts from the position defined by the *start* argument (*start* >= 1). The function resets to zero if no occurrence of the *key* argument is found.

> Example:
> search("Purchase order","order,1) = 10
> search("Purchase order","c",1) = 4

trim (*string*): Returns the converted *string* argument. All spaces encountered at the beginning and end of the string are deleted. The number of spaces included between two words is reduced to one.

> Example:
> trim(" Purchase order") = Purchase order

trimall (*string*): Returns the converted *string* argument. All spaces encountered are deleted.

> Example:
> trimall("Paris / New York / Rome") = Paris/NewYork/Rome

upper (*string*): Gives the *string* converted into upper case.

> Example:
> upper("Purchase order") = PURCHASE ORDER

ztrim *( ):* Deletes all zeros, starting from the left, in fields that are entirely numerical.

> Example:
> If a field named WEIGHT displays a value of 000200, then:
> ztrim(weight) = 200

Defining the properties of a Formula data source

Command: Data source > Formula > Properties...

1. Enter the formula directly in the **Edit** box.

- or -

Select the elements of your choice, then click **Insert.**

2. Click **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.

3. Click **OK.**

Hint: You can insert an element by double-clicking on it.

Note: If a variable used in the formula has a name containing one of the following characters &+-*/<>=^%,!\", it must be enclosed in brackets {}.

Note: You can check your formula by clicking Test. If message displays the formula value, it means that your formula is correct. If the value is not correct, follow the instructions displayed on the screen to carry out the necessary modifications.
If the value obtained is truncated, you must modify the maximum length specified in the Output tab.

Demonstration: Creating a simple formula

<u>Displaying the price of a product</u>
The production label must show the price of the product as a function of its weight and the price per kilogram.

1. Open a label. Two variables have to created: WEIGHT and PRICEPERKG.
2. For the WEIGHT variable: Enter 788 (the weight of the product being 788 g) as the Local value, and enter ''Please enter the weight in g'' in the Prefix box, then click OK.
3. For the PRICEPERKG variable: enter 15.70 (the price/kg being FF15.70) as the Local value of the variable, and enter ''Please enter the price per kg'' in the Prefix box, then click OK.
4. Add a formula and name it price.
5. Enter the formula WEIGHT*PRICEPERKG/1000, then click **OK**.
6. Save your document.

Demonstration: Adding the "Warning" Formula variable for displaying a warning message

In the following sequence, we will create a formula to display a warning message telling the user that the value of the Total_Weight shared variable exceeds 1,000kg.

If the weight value exceeds 1,000kg, the message ''Attention! Error! Total Weight exceeds maximum!'' will appear.

1. Open the label.
2. Create a formula and name it ''Warning''.
3. In the **Formula** dialog box, enter the following expression:
 if(Total_Weight>1000, ''Attention\n!Error!\n Total Weight exceeds maximum!'', "").
4. In the **Output** tab, enter 50 in **Maximum length** and click **OK**.
5. Position the variable as text within the label.

6. In the **Text** dialog box, select **Scalable** as the font and set its **Height** to 12.70 mm.
7. In the **Paragraph** tab, check the **Wordwrap** option then check **Centered** in Alignment.

Information on 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 conditional tests 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.

Practical Workshop: Creating a specific modulo

In this workshop we will convert the ''Customer_Code'' EAN8 barcode into a 2/5 Interleaved barcode using the "Formula_4_NewCustCode" Formula data source.

The barcode must have the following properties:

- Symbology: printer,
- Height: 4mm,
- Narrow bar width: 1mm,
- Ratio: 2,
- Human readable: below and centered,
- Distance from the bars: 0mm,
- Characters Font: printer font.

1. Open the ORDER_WS2.LAB label.

To calculate the weight

Create the Formula_1_Weighted formula, bearing in mind that the first character of Customer_Code is multiplied by 1, the second by 2, the third by 1, the fourth by 2, and so on.

The maximum output length of the variable is 6.

```
Formula_1_Weighted:
mid(Customer_Code,1,1)*1&mid(Customer_Code,2,1)*2&mid(Customer_Code,3,1)
1&mid(Customer_Code,4,1)*2
```

To add up the result of the weight calculation:

The next step involves adding together the figures resulting from the previous formula, bearing in mind that the maximum permitted length of this character string is 2.

Create a second formula and name it Formula_2_Sum.

To calculate the check digit:

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

Create a third formula and name it Formula_3_CheckDigit.

```
The expression is as follows:
if ((Formula_2_Sum % 10) > 0,10 - Formula_2_Sum % 10,0)
```

To calculate the data to be encoded:

When creating the barcode you must include the data to be encoded, for example, the value of the Customer_Code variable concatenated with the value of the check digit (Formula_3_CheckDigit).

Create a fourth formula and name it Formula_4_NewCustCode. This formula is the result of concatenating Customer_Code and Formula_3_CheckDigit.

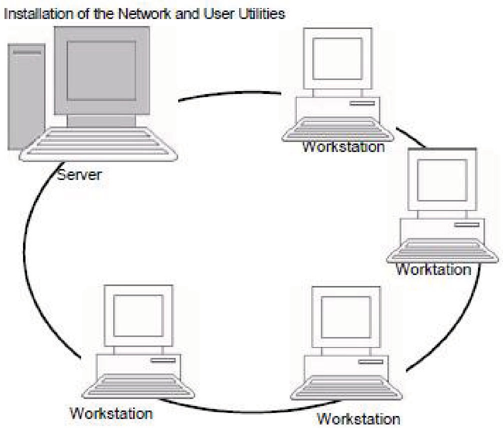
To create the barcode:

1. Select the Formula_4_NewCustCode formula then drag and drop it into the label over the Customer_Code barcode.
2. Define the properties of the barcode.

Installing the network version

Description

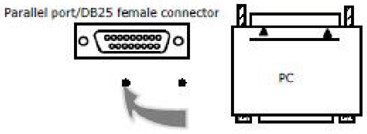
To use the network/multi-user version of Codesoft, you must first install Network and User Utilities either on the server or on a workstation that will act as the server, then install your labeling software on each workstation.

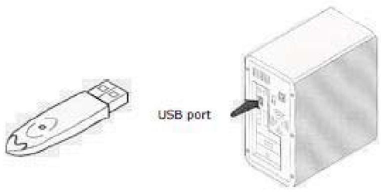


Installing the dongle

The dongle must be installed on the computer (server or workstation) on which the license manager is installed. A single dongle, pre-programmed with the number of licenses purchased, is included with the network version of your software.

The dongle must be installed before the license manager (License Manager or Service) can be started.





Note: When you start the program, if the dongle does not match the product version, a dialog box with the necessary instructions is displayed.

If you need to use a printer on the parallel port, connect it to the dongle. In this case, you might have to turn the printer on in order for the dongle to be recognized.

Installation procedure

Network configuration

Before you install the software, the network administrator must first define the structure of the network for the group of users, specifically:

- Define the license server on which the **Network and user Utilities** and dongle will be installed.
- Define the workstations, or the client workstations that will use the labeling software.

Description of Network Manager

The **Network and User Utilities** lets you use the network configuration of your labeling software. **Network Manager** includes:

- The License Manager (**License Service**).
- **Network Settings Wizard**: The **Network Settings Wizard** helps you define the network configuration.
- **User Manager**: The **User Manager** is installed with the **Network and User Utilities** so you can define access rights to the labeling software in a network setting.

Installing the Network and User Utilities on the server

Before installing the labeling software on all the workstations that will use it, you must install the Network and User Utilities on the server to configure the network.

1. Insert the CD for the installation in the appropriate drive.
   The Installation window is displayed.

   If the CD for the installation does not run automatically:
2. Go to **Start > Run**, then type the letter of the CD-ROM drive, followed by index.hta (for example, type D:\index.hta).
3. Select **Network and User Utilities**, which includes **License Manager**, **Service** and **User Manager**. Then click the **Install** button.
4. Select a type of protection : a software key (electronic code) or a hardware key (dongle).
5. Follow the instruction on the screen.
6. Share the [TKDONGLE] folder with full control, using TKDONGLE as the share name.

   The default access path for this folder is C:\Documents and Settings\All Users\ApplicationData\TKI\LicenseManager\TKDongle (Vista:C:\ProgramData\TKI\LicenseManager\TKDongle) > Right-click > Properties > Sharing tab and Permissions button.
7. If you want to define settings for your network configuration, start the **Network Settings Wizard** on the server. By default, if you do not modify the configuration, each workstation will have its own settings.

Note: For administrators:

Users wanting Write Access to the Network Licence must be dually given the rights by:

1. Sharing the TKDongle folder and authorizing the user: C:\Documents and Settings\All Users\Application Data\TKI\LicenseManager\ TKDongle (Vista: C:\Program Data\TKI\ LicenseManager\TKDongle) > Right-click > Properties > Sharing tab and Permissions button.

2. Give Write Access to the user in the Security tab of the TKDongle properties

Configuration

All the necessary tools to configure the network version are available from the ''Network'' tool bar accessible from by going to **Start > Programs > Network and User Utilities** and selecting **Network**.

The **Network Settings Wizard** helps you define the settings for your network version.

1. To start the **Network Settings Wizard**, click on the icon.
2. In step 1 of the wizard, select a settings mode: **generic**, **by user** or **by station**.

   - **Generic**: All users will use the same settings on all workstations. (user.ini).
   - **By user**: Each user can access his or her own settings on any workstation. (user name.ini).
   - **By station**: Each workstation has its own settings (station.ini).

3. In step 2, specify the location in which you want to store these settings. If you want to share these settings between various workstations, specify a network path accessible to all workstations.
4. In step 3, specify the location in which you want to store the shared data (variables, lists, printing logfile, etc.).

To configure the User Manager

If you want to define network access rights for all users of the labeling software, you must do so during installation (consult the User Manager help system).

- Click on the User Manager icon avialable on the Network toolbar.

Starting the Licence Manager

Before installing the labeling software on all workstations, you must start the License Manager.

If the License Manager was installed as **Service**, you do not need to start it. In fact, **Service** starts when the workstation is turned on and runs as a background task as long as the workstation is on. However, if you installed the **License Manager**, you must start it manually.

To start the License Manager

Click the icon available on the Network toolbar -or double-click the LICENSE.EXE file in the [DONGLE] folder in C:\PROGRAM FILES\TKI\9\COMMON\TOOLS\

Note: To start License Manager automatically when a Windows session is started, copy the shortcut for the program to the **Start > Programs > Startup** menu in Windows.

Installing the software on the workstations

The labeling software must be installed on all the workstations on which it will be used.

To install the software on a workstation

1. Insert the CD-ROM in the appropriate drive.
   The Installation window is displayed.

   If the CD-ROM does not run automatically:
2. Go to Start > Run, then type the letter of the CD-ROM drive, followed by index.hta (for example, type D:\index.hta).
3. Select the product to be installed, click the **Install** button and follow the instructions on the screen.
4. Start the labeling software. A message is displayed to inform you that no dongle has been found. Click **Yes** to start the software.
5. From the **Tools** menu, choose **Network Administration**.
6. Enable Use **Network License**.
7. Click **Modify** to select the server on which the license manager and dongle are installed.
   - or -
   Click **Browse** to automatically search for the server on which the license manager is installed.

   If the network has already been configured, a message asking if you want to use the current network configuration is displayed.
8. If you want to modify or configure the network settings, click the **Network Settings Wizard** button.
9. Click **OK**.
10. Restart the program.

United States	France	Germany	Singapore	China	Japan
1-414-228-3335	33-562-601-080	49-6103-30026-0	65-6477-7293	86-21-6100-6588	81-45-461-3603

www.teklynx.com

Microsoft GOLD CERTIFIED Partner