
LV
TEKLYNX
LABELVIEW
TEKLYNX
LABELVIEW
Barcode Better
powered by:
TEKLYNX
Barcode Better

用户指南

2026 版

本指南中的資訊不具備契約性，如有修改，恕不另行通知。

指南中所說明的軟體是按照授權協議進行販售。只有在授權協議條款許可的條件下，才可使用、複製或重製本軟體。

在沒有取得 TEKLYNX Corporation SAS 的書面同意時，不得以任何方式或為了達到任何目的（除了購買者個人使用之外）而複製、重製或傳輸本手冊中的任一部分。

©2026 TEKLYNX Corporation SAS,

版權所有。

目錄

關於本手冊 6

印刷慣例 6

關於產品 6

連接至資料庫 7

概念回顧 7

實際操作 1 8

導入數據 9

创建变量对象 10

Active Query Builder 11

查詢生成器 11

入門指南 11

新增物件到查詢 12

編輯物件屬性 13

加入資料表 14

為輸出欄位排序 15

定義準則 15

定義參數化查詢 16

查詢結果網格 16

數據庫管理器 18

資料庫的檔案結構 18

從連線清單中選擇一個資料庫 18

在資料庫內選擇一個表格 19

將表格新增至現用資料庫 19

刪除現用資料庫內的表格 20

檢視/ 隱藏現有表格的資料 20

定義一個密鑰欄 20
定義一個欄位的內容類型 20
定義一個欄位的最大容量 21
允許 Null 21
編輯資料庫視窗 21
根據內容選擇記錄 22
選擇所有相同的記錄 22
選擇一個相同的記錄 22
建立一筆新記錄 23
修改一筆記錄 23
刪除一筆記錄 23
資料庫查詢視窗 24
新增查詢 24
選擇/ 取消選擇一個或以上的欄位 24
修改選擇欄位的順序 25
使用預定義的資料建立一個篩選器器 25
將一個邏輯運算元套用至數個篩選器器 25
移除篩選器 25
修改 SQL 內的篩選器器 26
列印視窗 26
選擇一個要列印的文件 26
選擇一個現有的標籤範本 27
選擇印表機 28
公式 29
Formula（公式）資料來源 29
關於函數 29
運算元 30
算術運算元 30

比較運算元 30
串連運算元 31
邏輯運算元 31
檢查位元運算函數 31
轉換函數 35
ATA 2000 轉換函數 42
日期和時間函數 45
邏輯函數 55
數學函數 56
文字函數 61
定義 Formula（公式）資料來源的屬性 68
實際操作 - 計算一個特定「模組 Modulo」 68
安裝 Network 71
功能說明 71
安裝程序 72
開始安裝前 72
Network License Manager 說明 72
安裝 Network License Manager 72
組態設定 73
啟動 License Service 73
在工作站上安裝軟體 74
在工作站上安裝軟體 74

關於本手冊

印刷慣例

本手冊使用以下慣例來區分不同類型的資訊：

- 取自介面本身的詞彙會以粗體表示，例如指令。
- 會以全大寫字母顯示按鍵。例如：「按 SHIFT 鍵」。
- 已編號的清單表示需要遵循特定程序。
- 一段文字旁邊出現連接詞「- 或 -」時，表示可選擇另一個程序來完成要執行的作業。
- 一個功能表指令包含子功能表時，將以粗體顯示該功能表名稱及隨後應選擇的指令。因此，「請至 **File（檔案）> Open（開啟）**」就表示選擇 **File（檔案）**功能表，然後再選 **Open（開啟）**指令。

關於產品

您的產品可能無法使用本手冊中所說明的部份功能。

如需取得軟體提供特定功能的完整清單，請參閱產品隨附的規格表。

連接至資料庫.

概念回顧

本章可讓您真正見識本軟體的功能有多強大。現在，我們要 利用 ODBC（開放式資料庫連結）和 OLE DB（物件連結與 內嵌資料庫）連線將您的標籤（容器）與資料庫（內容）連接起來。

資料庫

資料庫可讓您儲存資料，所有資料將被整理成雙維式表格，稱之為關係。表格中的每一列都被稱為一個記錄。記錄的功 用在於管理物件，其內容會以欄位形式被整理遍佈至表格的 不同欄 內。一個 資料庫內 可能包含數個表格。要將一個已知資料庫內的 同表格串連起來，我們必須使用連接。本章稍後將介紹一 個實例來示範如何建立連接。

ODBC

這是資料庫存取標準。ODBC 可提供一個直接連接應用程式的方式，例如將您的標籤設計軟體連接至幾個不同的資料庫。

本軟體提供數種 ODBC 驅動程式，讓您存取最新的資料庫。這些驅動程式如下：

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

OLE DB

這是一個連線標準，用於存取所有資料庫標準及訊息系統中儲存的資料。

實際操作 1

安裝 ODBC 資料來源及匯入資料

在數據能夠被訪問之前，第一步必須先安裝必需的數據源。

安裝 ODBC 數據源

下面的一些步驟是採用直接創建模式. 如果您想以向導模式創建，請在數據庫單擊鼠標右選 擇精 灵來 進行.

連接到 TKTraining.mdb 數據庫:

1. **選擇 工具** > **管理員 ODBC**.

2. 在 **User DSN**（用户 **DNS**）選項卡中

注意: 您可以定義數據源為系統數據源名稱 (DSNs). 這 些數據源對於特 定 的 電 腦來說是唯一的，而 對 於特定用戶非唯一性. 任何有權限的用戶都可以訪問此 DSN.

3. 選擇 **Microsoft Access** 驅動器（*.mdb）。單擊 **Finish** （完成）.

4. 在 **Data source name**（數據源名稱）框中輸入一¥'¥'TK Training CS Level 2¥'¥' .

5. 點擊 選擇 並選擇到數據庫文件 TKTraining.mdb, 此文件位於 InstallDir¥Samples¥Forms¥Tutorial.

6. 點擊 選項 按鈕. 勾選 隻讀 屬性. 此選項可以避免在 打開數據庫文件時對數據庫文件有任何的讀/寫操作.

7. 点击 确定 来退出 ODBC 连接对话框.

導入數據

安裝了數據源, 我們現在可以從數據庫中導入數據並將它 插入標簽中.

1. 打開標簽文件 PRODUCT_WS3.lab.

2. 選擇 **Data sources**（資料庫來源）> **Database**（資料庫）> **Create/Edit Query** （建立/編輯查詢）...。

3. 從 **Select a data source** （選擇資料來源）列表中選擇 TK Training Level 2.

4. 選擇 ¥"Fruits¥" 字段在 **選擇表格** 列表處. 此時數據庫字段應該顯示在 **選擇字段** 列表處

5. 分別選擇字段 "ProdName", "Origin", "Weight" 以及 "Reference" .

6. 點擊按鈕  . 此按鈕可以將數據庫記錄按照字母或數字進行升序或降序排序

7. 作為 **排序關鍵字** 並將 ¥"升序¥" 作為 **排序順序**.

8. 保存查詢 PRODUCT_WS4_ODBC.CSQ.

9. 單擊 OK （確認）。

   變量在 Document Browser（文檔瀏覽器）的 Data Sources（數據源）選項卡的 Database （資料庫）目錄中列出。

*文件在 InstallDir¥Samples¥Forms¥Tutorial

來瀏覽或打印不同的數據庫記錄， 請使用屏幕上方的導航條. 您也可以從 查看查詢結果數據 窗口來打印數據.



创建变量对象

1. 從 數據庫 字段列表中選擇所需要創建的變量, 然后把它拖到標簽設計區域鬆開鼠標左鍵
2. 在彈出的菜單裡面選擇 文本.

Active Query Builder

查詢生成器

查詢生成器可以幫助您通過可視化的界面來完成復雜 SQL 查詢語句的構建工作.

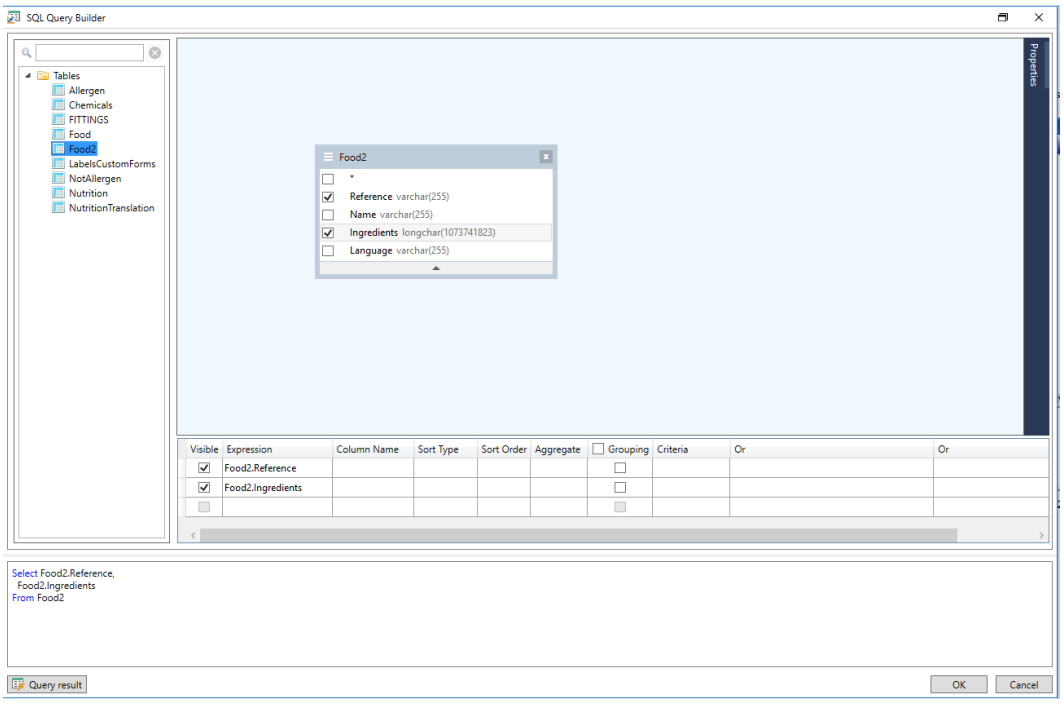
要使用查詢生成器, 您必須有最基本的關於 SQL 方面的知識. 查詢生成器將幫助您來創建正確的 SQL 語句, 了解最基本的 SQL 語句將使您事半功倍..

入門指南

這就是「活動查詢產生器」啟動時的外觀。

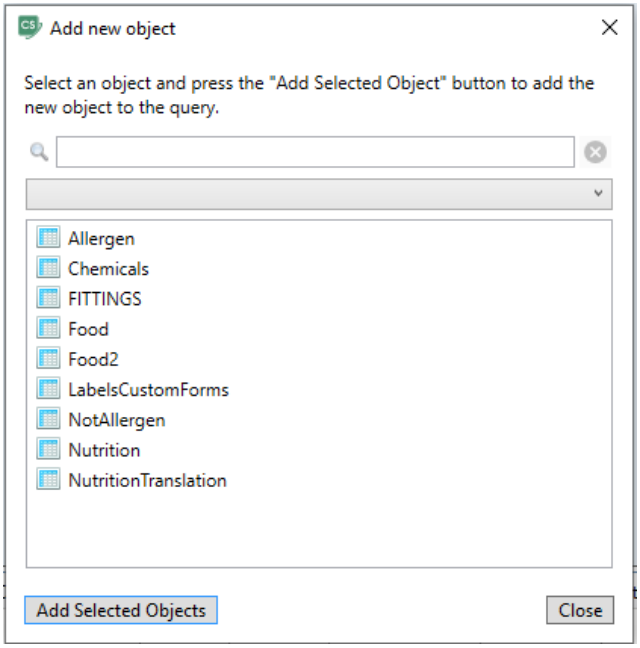
主視窗可分成以下部分：

- **查詢建立區域**是顯示查詢視覺表示法的主要區域。該區域可讓您定義來源資料庫物件和衍生資料表以及它們之間的連結，並設定資料表和連結的屬性。
- 欄窗格位於查詢建立區域下方。透過使用查詢輸出欄和運算式，可用來執行所有必需操作。在此您可定義欄位別名、排序和群組，以及定義準則。
- **表格視圖區域**位於左側。您可以在此處瀏覽查詢，並快速找到它的任何部分。使用窗格頂部的**搜尋**欄位來指定搜索。
- 查詢建立區域上方的頁面控制可讓您在主查詢和子查詢之間切換。
- 查詢建立區域一角標有字母「Q」的小區域是聯集子查詢處理控制項。在此您可新增聯集子查詢並執行與其有關的所有必要操作。



新增物件到查詢

要新增物件到查詢，用滑鼠右鍵按一下查詢建立區域，然後從下拉式功能表選擇「新增物件」項目。

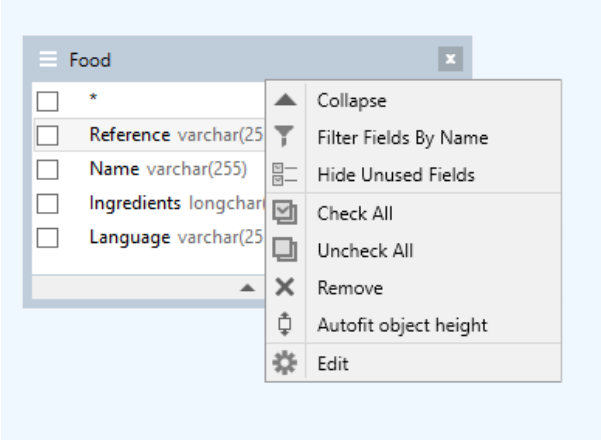


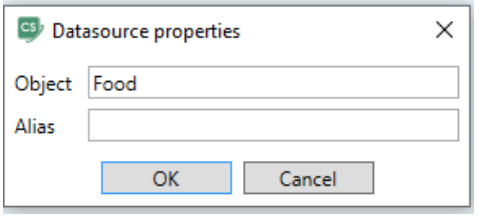
新增新物件視窗允許您一次新增多個物件。表格元素將列在下拉式列示方塊中。**顯示所有物件**篩選器將顯示所有可供選擇的物件。位於視窗頂部的**搜索欄位**用於尋找所需的對象。您可以透過按住 **CTRL 鍵**選擇一個或多個物件，然後按**新增選定物件**按鈕，將這些物件新增到查詢中。您可以重複執行此操作多次。新增完物件後，按一下**關閉**按鈕以隱藏此視窗。

要從查詢中刪除物件，請按一下物件標頭中的**關閉**按鈕。

編輯物件屬性

透過在物件上按一下滑鼠右鍵並從下拉式功能表選擇編輯...項目，或只需連按兩下物件標題，即可變更新增至查詢的每個物件的屬性。

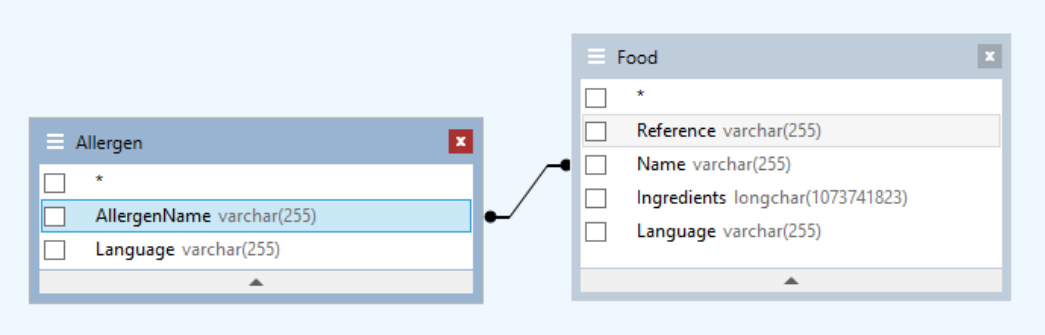




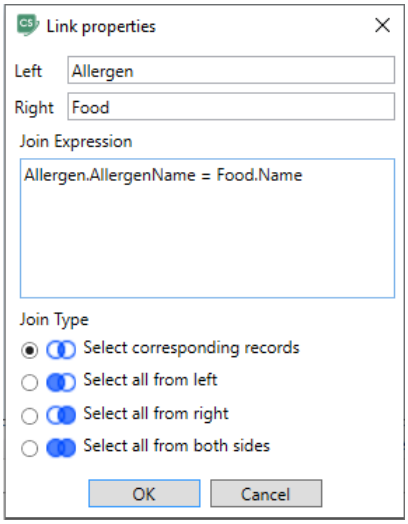
資料來源屬性對話方塊視乎不同的伺服器而異，但至少所有資料庫伺服器的「別名」屬性相同。

加入資料表

要在兩個物件之間建立連結（即加入它們），您應選擇要將其連結到另一物件的物件欄位，並將其拖曳到另一個物件的相應欄位。完成拖曳後，會出現一條連接已連結欄位的直線。



預設情況下，所建立的加入類型是「內部加入」，即僅兩個資料表中相符的記錄才會包含在結果資料集內。要定義其他加入類型，您應使用滑鼠右鍵按一下連結並從下拉式功能表選擇編輯...項目，或對其連按兩下打開連結屬性對話方塊。此對話方塊可讓您定義加入類型和其他連結屬性。



要刪除物件之間的連結，請按右鍵連結線，然後選擇下拉式功能表中的**刪除**選項，或者選擇連結線並按**刪除**按鈕。

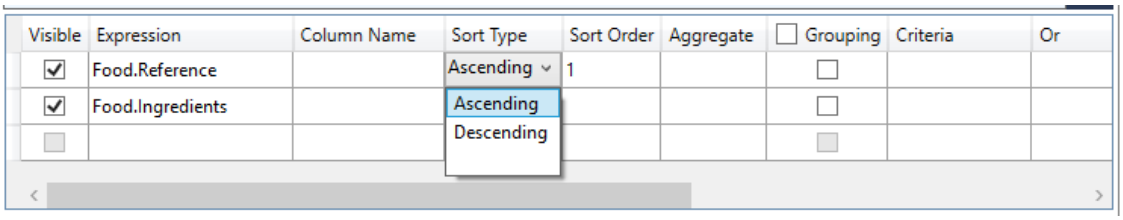
為輸出欄位排序

要啟用輸出查詢欄位排序，您應使用**欄窗格**的**排序類型**和**排序順序欄**。

排序類型欄可讓您指定欄位的排序方式：按**遞增**還是**遞減**順序。

如要排序多個欄位，**排序順序**欄可讓您設定欄位的排序順序。

要停用某些欄位的排序功能，您應清除該欄位的**排序類型**。



定義準則

要定義在**欄窗格**列示的運算式準則，您必須使用**準則欄**。

在該欄中，您應輸入省略運算式本身的準則。在查詢中找到以下準則

WHERE（欄位 >= 10）AND（欄位 <= 20）

您應將

>= 10 AND <= 20 寫入準則欄。

您也可以使用**條件**列中的**下拉清單**，以將常用運算式和運算子貼到編輯欄位中。對於進階條件建立，請使用**條件**列中的**運算式編輯器**按鈕。**運算式編輯器**對話方塊即會顯示。

您可使用 Or... 欄為一個運算式指定多個準則。可使用 OR 運算子將這些準則在查詢中串聯起來。

定義參數化查詢

查詢產生器可讓您建立參數化查詢，其中參數值包含於變數之中。

附注：您必須先建立變數。

1. 拖曳並釋放要在其中執行查詢的資料表。
2. 選擇要在其中套用準則的欄位。
3. 在準則欄或 SQL 格式編輯欄位中，指定用作搜尋準則物件的變數。

範例：要查找您預先建立的變數 Var0 的值：

- **在 SQL 中：**

  SELECT [Table].*

  FROM [Table]

  WHERE [Table].Field = APPLICATION.DOCUMENT.Var0

- **準則欄**

  = APPLICATION.DOCUMENT.Var0

4. 按一下**查詢結果**按鈕以顯示查詢的結果。

查詢結果網格

如要進入**查詢結果** 網格，應點選**定義查詢 對話框**內的  按鈕 、 **合併資料庫瀏覽器** 工具列，或透過 **資料來源** > **資料庫** 功能表進入

此網格可顯示一個查詢的結果，或搜尋特定詞語或其所有出現的位置，以及列印相關的標籤。

注意：針對包含 **BLOB 資料類型**（例如，影像檔）的資料庫欄位，「**<Binary data>**」標記將在「查詢結果」對話方塊中顯示。

查詢結果 網格包含：

- **搜尋函數** 

  搜尋欄位，可輸入將要執行搜尋的欄位

  要搜尋的資料，可輸入要搜尋的數值

  在欄位內任何位置或 欄位開端搜尋數值。

- **導覽功能，可瀏覽查詢結果記錄**

  第一筆記錄 

  上一筆記錄 

  下一筆記錄 

  最後一筆記錄 

- **結果網格**

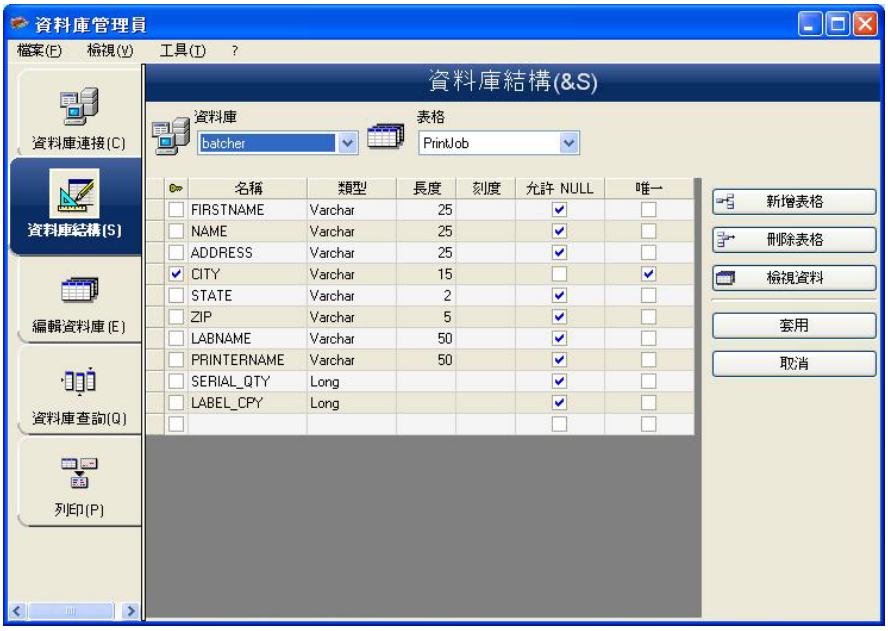
  顯示查詢結果的出現記錄。

- **重新詢問**

  重新詢問要求並更新資料表。

數據庫管理器.

資料庫的檔案結構



資料庫結構視窗用於管理資料庫檔案的結構：新增、修改或刪除表格／欄位等。

從連線清單中選擇一個資料庫

1. 點選 **Database** （資料庫）下拉式清單.
2. 點選所要的資料..

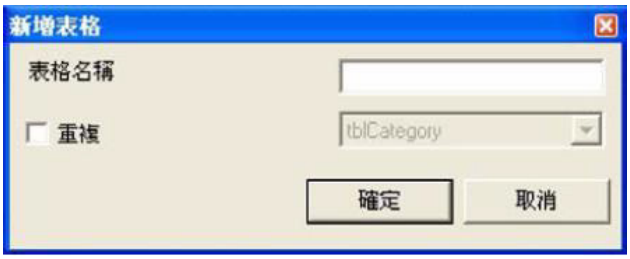


在資料庫內選擇一個表格

1. 點選 **Table** （表格）下拉式清單.
2. 點選所要的資料.

將表格新增至現用資料庫

1. 點選 **Add table** （新增表格）.
2. 輸入新表格的名稱.
3. 點選 **OK** （確定）.



您也可以從選取的資料庫中複製一個現有表格的結構。如要這麼做：

1. 勾選 **Duplicate with** （重複）旁的方框.
2. 點選下拉式清單.
3. 點選所要的資料.
4. 點選 **OK** （確定）.



刪除現用資料庫內的表格

1. 點選 **Table** （表格）下拉式清單.
2. 點選所要的資料.
3. 點選 **Delete table** （刪除表格）.

檢視/ 隱藏現有表格的資料

1. 點選 **View data** （檢視資料）.

定義一個密鑰欄

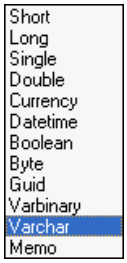
1. 勾選所要欄位旁的方塊.



2. 點選 **Apply** （套用）.

定義一個欄位的內容類型

1. 在 **Type** （類型）欄內點選所要的欄位.
2. 點選下拉式清單按鈕.
3. 點選所要的資料.



4. 點選 **Apply** （套用）.

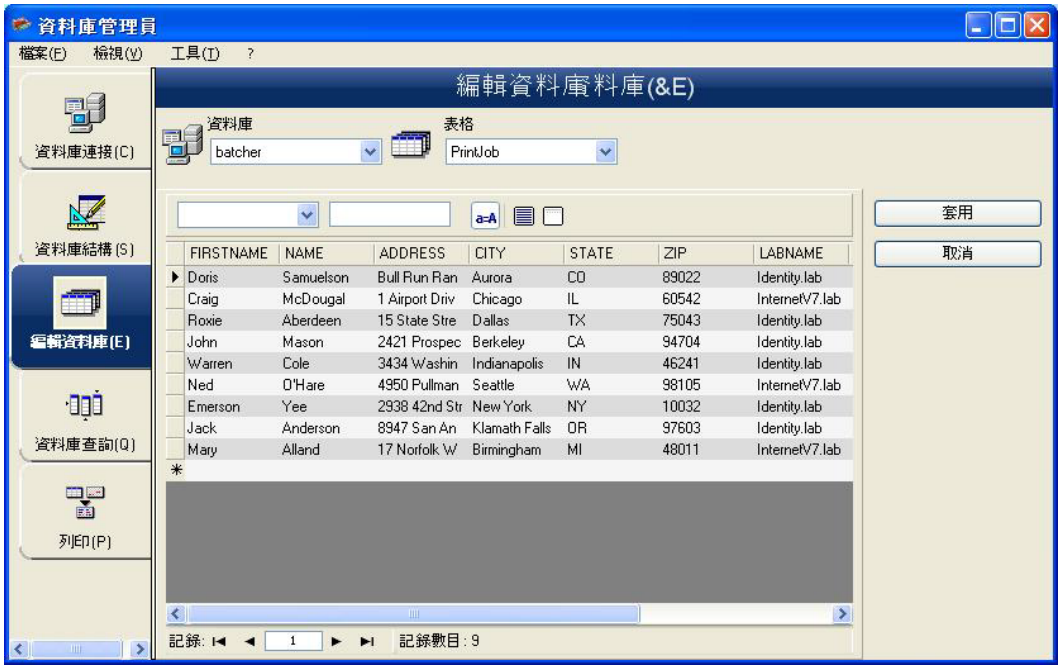
定義一個欄位的最大容量

1. 在 **Length** （長度）欄內點選所要的欄位.
2. 輸入所要的數值.
3. 點選 **Apply** （套用）.

允許 Null

1. 為所要的欄位勾選 **Allow Null** （允許 **Null**）方格.
2. 點選 **Apply** （套用）.

編輯資料庫視窗



編輯資料庫視窗用於管理資料庫檔案的內容：新增、修改或刪除資料。

這些動作須取決於資料庫的類型。因此， Excel 檔案記錄是無法修改的。

根據內容選擇記錄

利用欄位的內容尋找一筆記錄.

1. 點選下拉式清單按鈕.
2. 點選所要的資料.
3. 點選資料輸入欄.
4. 在資料輸入欄內輸入所要的數值

選擇所有相同的記錄

已找到至少一筆記錄。

1. 點選下拉式清單按鈕.
2. 點選所要的資料.
3. 點選資料輸入欄.
4. 在資料輸入欄內輸入所要的數值.
5. 點選 **Select all** （全選）按鈕（）

選擇一個相同的記錄

並已找到至少一筆記錄。在搜尋欄位內必須有幾筆相同的內容。

如要選擇一筆記錄，應使用搜尋工具：點選 1 （第一筆）、2 （上一筆）、3 （下一筆）或 4 （最後一筆）。



建立一筆新記錄

1. 在標示星號的列中點選一個欄位.
2. 在相應的欄位內輸入所要的數值.
3. 點選 **Apply** （套用）.

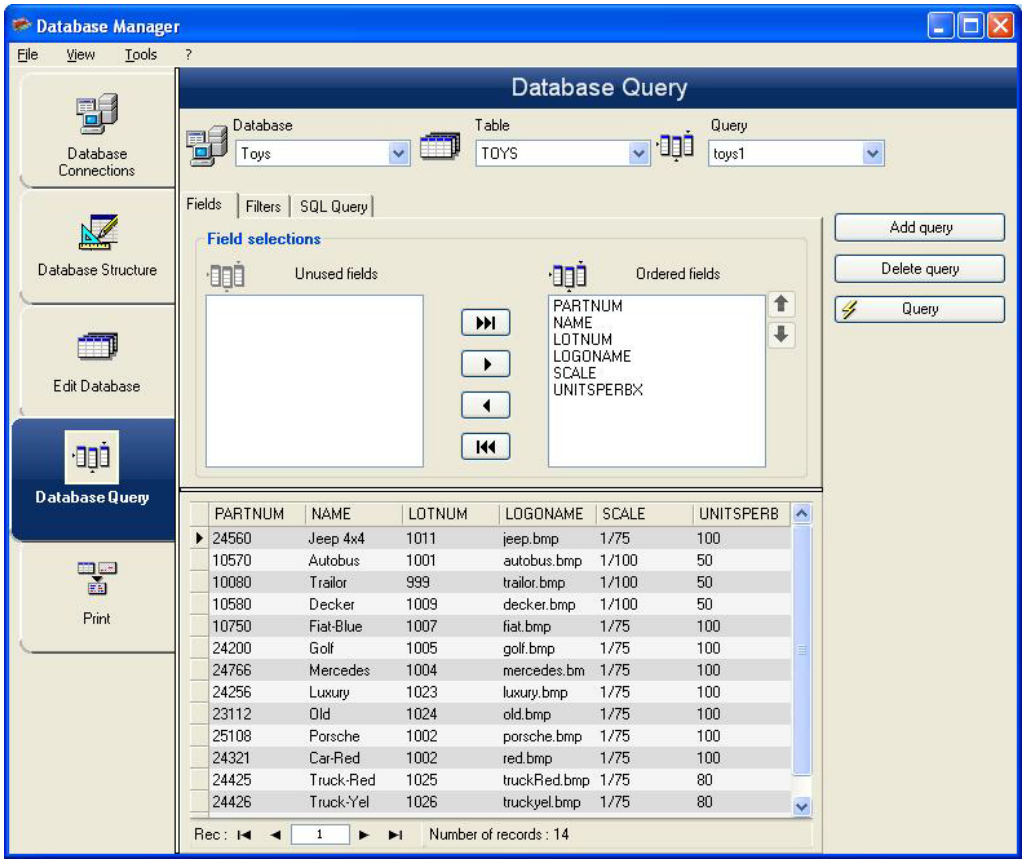
修改一筆記錄

1. 點選您想要修改的資料.
2. 輸入所要的資料.
3. 點選 **Apply** （套用）.

刪除一筆記錄

1. 點選所要欄位的資料庫游標.
2. 按滑鼠右鍵點選所要欄位的資料庫游標.
3. 點選背景功能表內的 **Delete Record** （刪除選取的記錄）.

資料庫查詢視窗



資料庫查詢視窗用於建立及套用各種篩選器器。

新增查詢

1. 點選 **Add query** （新增查詢）.
2. 為查詢輸入一個名稱.
3. 點選 **OK** （確定）.

選擇/ 取消選擇一個或以上的欄位

1. 在導覽工具內點選  .
2. 點選 **Query** （查詢）.

修改選擇欄位的順序

1. 點選 **Ordered fields** （已排的欄位）視窗內的所要欄位.
2. 點選上或下箭頭移至所要的資料.
3. 點選 **Query** （查詢）.

使用預定義的資料建立一個篩選器器

1. 點選 **Filters tab**.
2. 點選 **Add row** （新增列）按鈕（）.
3. 點選 **Field** （欄位）欄點選下拉式清單按鈕點選所要的資料.
4. 點選 **Operator** （運算元）欄點選下拉式清單按鈕點選所要的數值.
5. 點選 **Value** （數值）欄輸入所要的數值.
6. 點選 **Query** （查詢）.

將一個邏輯運算元套用至數個篩選器器

1. 點選 **Add row** （新增列）按鈕 ().
2. 點選 **Logical** （邏輯）欄點選下拉式清單按鈕點選所要的資料 (AND or OR).
3. 创建篩選器.
4. 點選 **Query** （查詢）套用及檢視所做的變更。.

移除篩選器

注意: 必須存在至少一個篩選器器。

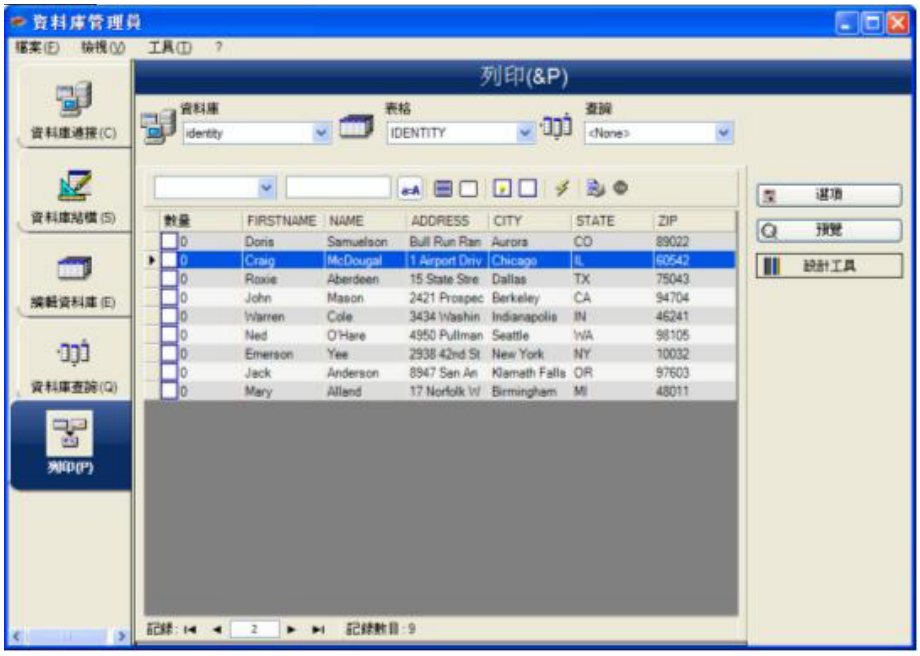
1. 點選所要欄位的資料庫游標.
2. 2. 點選 **Remove row** （刪除列）按鈕().

修改 SQL 內的篩選器器

注意: 必須存在至少一個篩選器。

1. 选择 **SQL Query** 标签.
2. 勾選 **Modify the query in SQL language** （用 SQL 語言修改查詢）.
3. 點選 **Query** （查詢）.

列印視窗



列印視窗用於選擇列印的檔案、指定印表機以及在列印前定義各項參數。

選擇一個要列印的文件

1. 從文件中選一個文檔.

2. 檢查文件名.

   -或者 -

1. 點選 **Create labels wizard** (建立標籤精靈）按鈕 ().
2. 按照精靈的指示操作.

注意: 建立一個與資料庫相關的標籤， 可讓您定義設置各個資料庫欄位所需的元素。

選擇一個現有的標籤範本

1. 點選 **Open an existing document** （開啟現有的文件）按鈕（).
2. 選擇一個.lab 檔案.
3. 點選 **OK** （確定）.

注意: Label name （標籤名稱）和 Printer name （印表機名稱）選項群組中的 Field（欄位）單選按鈕可讓您選擇所要的標籤或印表機， 後者由現用資料庫的 其中一個欄位定義。

從一個字段來選擇文檔

如果您的數據庫中包含需要打印的標籤名的字段, 您可以定義這個字段作為標籤名字段， 讓數據庫管理器來通過這個字段抓取所需要打印的.lab 文件.

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

在本例裡, 字段'Labname' 就可以作為包含標籤名稱的字段.

1. 檢查標籤名稱組.
2. 選擇所需要的字段.

選擇印表機

點選 **Add or remove a printer** （新增或移除印表機）按鈕.

公式

Formula（公式）資料來源

指令：**Data source（資料來源）> Formula（公式）> Add（新增）**

Formula（公式）資料來源包含您所建立的資料來源清單。這些資料來源是由運算元、常數、資料來源、控制變數、公式以及函數等組合所產生。資料可為數字或是英數字。

為了在文件內執行計算，必須先建立 **Formula（公式）**資料來源。

此資料來源有特定的對話方塊， 讓您定義特定公式的必要函數。

關於函數

注意：Windows 系統區域設定可能會影響某些公式的輸出。

函數是預先定義的公式，使用稱為引數。

函數可用來求回一個數字、字串或邏輯數值、計算或運算的結果。

在公式定義方面共有組函數公式：

- 檢查位元運算函數
- 轉換函數
    - ATA 2000 轉換函數
- 日期和時間函數
- 邏輯函數

- 數學函數
- 字串函數

運算元

本程式包含算術、比較、串接及邏輯運算元。

算術運算元

運算元	功用
*	將兩個數目相乘
+	將兩個數目相加
-	從一個數目減去另一個數目，或將負數值指定至一個運算域
/	由一個數目除去另一個數目
^	增加一個數字的冪次方
%	模數

注意：某些通常預期會回傳錯誤的公式並不會回傳錯誤。例如：1/0、0/0、0.0/0.0、1/000 和 0^(-2)。這是有意為之的 以免阻塞列印過程。如果需要，使用者需要透過新增額外的控制項 來定義所需的行為。

比較運算元

運算元	意義
<	小於
<=	小於或等於
>	大於
>=	大於或等於

=	等於
<>	差異

串連運算元

用來結合兩個字串。

運算元	意義
&	串連兩個字串

邏輯運算元

（亦可參考邏輯函數）

運算元	意義
!	非邏輯

檢查位元運算函數

注意：Windows 系統區域設定可能會影響某些公式的輸出。

addmodulo10 (string)：可求回整數，末端附加一個 Modulo 10 檢查字元。

addmodulo10_212(string)：可求回一個模數 10_212 的檢查字元（標準模數 10 檢查字元的變體），並包含當前字串及字串末端的檢驗和。

addmodulo43 (string)：可求回整數，末端附加一個 Modulo 43 檢查字元。

bimodulo 11 (string)：可求出兩個模數 11 的檢查字元（Code 11）。

canadacustomscd (string)：可求回加拿大海關標準（Canada Customs Standard) 檢查字元。

Check103(string): 此功能函數會計算 Mod103。

check 128 (string)：可求回 Code 128 和 EAN-128 的檢查字元。

checkPZN(string): 計算 PZN 條碼的檢查數字。PZN 條碼是依據德國製藥產品使用的 Code 39。

CheckSum2Mod47(string): 此功能函數會計算 Mod47。

CheckSumMod10(string): 此功能函數會計算 Mod10。

CheckSumMod10_Codabar(string): 此功能函數會計算 Codabar 的 Mod10 總和檢查碼。

CheckSumMod10_MSI(string): 此功能函數會計算 MSI 條碼 的 Mod10 總和檢查碼。

CheckSumMod11_3Suisse(string): 此功能函數會計算 3Suisses 的 Mod11 總和檢查碼。

CheckSumMod34(string): 此功能函數會計算 Mod34。

checkupce (*string*)：可求回 UPC-E 的檢查字元（6 個必要字元）

CRC16(string): 傳回 CRC-16 值。

例如：

CRC16("123456789") = 47933

"0x" & hex(CRC16("123456789"), 4) = 0xBB3D

modulo 10 (value)：可求回 Code 2 of 5 interlaced、EAN-13、EAN-8、EAN-128 K-Mart、VPCA 的模數 10 檢查字元。

modulo10_212(string) 可求回模數 10 的檢查字元變體

modulo10IBM(string)：可求回 IBM 模數 10 檢查字元（標準模數 10 檢查字元的變體）

modulo10UPS(string)：可求回 UPS 模數 10 檢查字元（標準模數 10 檢查字元的變體）

modulo 11 (string)：可求出模數 11（code 11）的檢查字元。

modulo11IBM(string)：可求回 IBM 模數 11 檢查字元（標準模數 11 檢查字元的變體）

modulo16(string)：可求出模數 16 的檢查字元。

modulo 24 (string)：可求出模數 24（Code PSA）的檢查字元。

modulo 32 (value)：可求出模數 32（義大利製藥業）的檢查字元。

modulo 43 (string)：可求出模數 43（code 39）的檢查字元。

modulo 47 (string)：可求出兩個模數 47（Code 93）的檢查字元。

Plessey (string)：可求出兩個檢查字元（Plessey Code）。

pricecd (*string*)：可求回 4 個 UPC Random Price 檢查字元。

pricecd5 (string)：可求回 5 個 UPC Random Price 檢查字元。

StringToExt39(string): 此功能函數會準備在 Extended 39 條碼 使用的字串。

UPSCheckDigit (string): 傳回 UPS 檢查字元。此方法的輸入是一個字串。此方法會讓您決定追蹤號碼的其餘部分。

它採用 15 個字元的序列，並使用這個序列計算出檢查號碼。

1 - 頭兩個字元必須是 "1Z"。

2 - 後接的 6 個字元則應填入我們的 UPS 帳戶號碼 "XXXXXX"。

3- 後接的 2 個字元表示服務類型：

- "01" 是次日送達空運。

- "02" 是隔天送達空運。

- "03" 是陸運。

4- 後接的 5 個字元是我們的發票號碼 (我們的發票有 6 位數，但會去除第一個數字。例如，123456 發票會成為 23456 的字元)。

5- 後接的 2 個數字是以 0 填滿的包裹號碼。例如，包裹 1 是 "01"，包裹 2 則是 "02"。

6- 最後和最終的字元是檢查號碼。

注意：上述序列提供 17 個字元，但只需要 15 個字元即可計算檢查號碼。若要計算檢查號碼，請移除 "1Z" 部分，並只在方法中使用後面的 15 個字元。

轉換函數

注意：Windows 系統區域設定可能會影響某些公式的輸出。

AI253(string)**:** 準備應用程式識別子 253 字串的特定功能函數。

AI8003(string): 準備應用程式識別子 8003 字串的特定功能函數。

ASCII(string)**：**會求回字串引數中第一個字元的 ASCII 碼。

例如：

ascii("A") = 65

char (integer):

- 從 128 到 255 的值：提供符合 ASCII 表中整數引數的字元。
- 所有負值和從 0 到 127 以及大於 255 的值：傳回符合整數引數的 Unicode 字元。

例如：

char (65) = "A"

CodabarData(data, start, stop)**:** 調整 CodabarData 的資料。

Code128CData(string): 調整 Code 128 C 的資料。

Currencytoeuro (value) **：**可將當前國家的貨幣換算成歐元。

DatabarData(data, min, max, hasComposite): 調整 DatabarData 的資料。

DBCSToUnicode(string,codepage) The formula takes two parameters - the data «string» to be converted, and the Codepage «codepage» used. The Codepage parameter is represented by the language name (Thai, Japanese, ChineseGBK, Korean, ChineseBig5, European, Eastern, Cyrillic,

Greek, Turkish, Hebrew, Arabic, Baltic, Vietnamese, UTF-8, ACP) or numeric value (search code page identifiers documentation on Internet).

例如:

DBCSToUnicode(unicodetoDBCS("傍傍傍傍", "UTF-8"), "UTF-8") =傍傍傍傍

This function is needed for data that come from non-Unicode file or data sources (e.g. Database)

Note: This function has reverse action to UnicodeToDBCS method.

dollar(value)：將一個欄位轉換成代表價格的欄位。

例如：

如果一個名為 PRICE 的欄位顯示 199 的數值，則：

dollar(PRICE) 將顯示 $199.00。

Ean128Data(string_1,string_2): Adjust "string_1" data for EAN 128 barcode according to template "string_2"

Eurotocurrency(value) :可將歐元換算成本國貨幣。

注意：您必須在公式 變數的 輸出 標籤內勾選 顯示小數 框，以定義要顯示的小數位數。使用的兌換率是在 選項 對話框的 其它標籤內定義的。

FileToData(fileName, errorData, maxSize): 傳回從檔案讀取的資料。

fixed (value , num_decimals, non_sep)可求得一個格式化後的字元串，其小數位數相等於 num_decimals 且含有或不含千位數分隔符。。

例如：

fixed (1234.5678, 3, TRUE) = "1234.568"

fixed (1234.5678, 3, FALSE) = "1,234.567"

注意：您必須在公式 變數的 輸出 標籤內勾選 顯示小數框,

以定義要顯示的千位數分隔符。

FormatDate(date, dateFormat, localeID): 轉換 «date» 為依照 «dateFormat» 格式所建立的字串。

返回值	字串	字串格式的日期值， 已根據格式進行格式化.
日期	字串	字串格式的日期值.
dateFormat	字串 / 數字	基準日期值的格式。 - 由數字編碼的預先定義格式。(例如， 1-"dd/mm"、2-"dd mmmm" 等,詳細見下文表) . 可能的值可在這裡找到。 - 0 或負值表示使用目前的系統日期格式 - 由字串值直接編碼的格式。(例如 "yyyy/mm/dd")
LocaleID	數字	參數可指定用來顯示日期值的地區設定。localeID 為一選用參數（預設值： 0， 會使用系統的地區設定）。 您可以在 http://msdn.microsoft.com/en-us/goglobal/bb964664.aspx 中找到可能的值（數值）

FormatMoney(amount, use separation characters(T/F), price characters, force leading zaro (T/F), location for price character, # of decimal places)

此功能函數允許你新增貨幣符號、點、強制前置零和使用分隔字元

例如：

FormatMoney("123456.234", "T","$","F",0,2) = $123,456.23

FormatMoney("1234","F","$","T",8,2) = $1234.00

FormatMoney("0.234","F","$","F",0,2) = $.23

FormatMoney("0.234","F","$","T",0,2) = $0.23

註：價格字符定位負責價格字符部位，在價格字符與數量之間添加空格。如小於結果字符串的長度，該值將被忽略。

FormatNumber(number): 此功能函數在存在值且零 (0) 代表始終顯示時，允許您在數值欄位中的井字 (#) 符號僅代表顯示時，進行格式化。

例如：

FormatNumber(123.45, "US$ #,###,###.00") = US$ 123.45
FormatNumber(123.45, "US$ 0,000,000.00") = US$ 0,000,123.45
FormatNumber(.45, "#,##0.00") = 0.45
FormatNumber(.45, "#,###.00") = 45
FormatNumber(7188302335, "(###) ###-####") = (718) 830-2335
FormatNumber(123.45, "00.00") = 23.45
FormatNumber(123.567, "###,##0.00") = 123.57

GetEnv(名稱)：傳回環境變數的 «名稱»（字串）值。

例如：

GetEnv("OS") = " Windows_NT"。

注意：此功能與使用者定義變數搭配使用

GS1AIData(AIName, AIData, isLastAI): 調整 GS1 AI 的資料。

GS1HRData(AIName, AIData, calcCheckDigit)**:** 調整 GS1 Human readable 的資料。

GS1Norm(string)**:** 調整 GS1 Norm 的資料。

int(value)： 可求回低於或等於 value 引數的最大整數。

例如：

int (-5.863) = -6

int (5.863) = 5

LmChar(整數)：返回對應于 ANSI 表中整數參數的字符。該表不依賴於本地系統。
值為 0 至 255 之間。

MaxiCodeData(Mode, PostalCode, CountryCode, ClassOfService, TrackingNumber, UpsShipperNumber, JulianDate, ShipmentID, PackageNumber, TotalPackages, PackageWeight, AdressValidation, ShipToAdress, ShipToCity, ShipToState)**:** 從 «string» 輸入中建立 Maxicode 資料。

text(value, format) ：可求回以 format 引數採用的格式編寫的 value 引數， 對應至表單變數的說明中敘述的格式。

例如：

text(012.345, "######,(###)") = 12,(345)

text(1012.345, "######,(###)") = 1012,(345)

trunc(value)： 可求回 value 引數的完整部份。

例如：

trunc (123.45) = 123

unicodetoDBCS(Data, "Codepage")：此公式使用 2 個參數 - 要轉換的資料以及使用的代碼頁 (code page)。代碼頁的參數可有下列數值：

泰文 (Thai)

日文 (Japanese)

中文 (Chinese GBK)

韓文 (Korean)

中文 Big5 (Chinese Big5)

東歐文 (European eastern)

希臘文 (Greek)

土耳其文 (Turkish)

希伯來文 (Hebrew)

阿拉伯文 (Arabic)

巴爾的文 (Baltic)

越南文 (Vietnamese)

例如：

unicodetoDBCS (**1234 中文測試 ABCD","Chinese Big5**") = **1234 中文測試 ABCD**

unicodetoDBCS (**1234 中文**测试 **ABCD","Chinese GBK**") = **1234 中文**测试 **ABCD**

注意： 代碼頁 參數無大小寫之分。須用英文述寫。

value(string) **：**可求得一個字串的數字數值。

例如：

value("123") = 123

value("0.00:48:00")-value("12:00:00") 等於 "16:48:00"-"12:00:00" 等於 0.00，4 小時 48 分鐘的序號。

注意：通常在公式內不需要使用 value 函數，因為程式會自動視需要將文字轉換成數字。

ValueEx(string)：轉換 «string» 為數值。

VoiceCode(string, string, string): VoiceCode 是使用 GTIN、Lot 及選擇性的 PTI 日期所計算出的 4 位數號碼。

Return value	string	計算 VoiceCode 值。
GTIN	string	14 位數 GTIN 號碼。只允許數字。在非數字值的情況下會傳回錯誤。在資料長度超過 14 的情況下會接受左側的 14 個位數。在資料長度比 14 短的情況下會在左側加上 '0'。
LotNumber	string	多達 20 個英數符號資料。在資料長度超過 20 位數的情況下會接受左側的 20 個符號。
Date	string	這是選擇性的參數。以 YYMMDD 格式代表日期的 6 位數資料。 公式在錯誤長度、非數字值或錯誤日期的情況下會傳回錯誤。

此計算會以下列方式執行：

1. 計算 PlainText。

   a. PlainText 是附加 Lot 碼和所在日期的 14 位數 GTIN。

   b. 不包含應用程式識別碼首碼或括號。

   c. 在 GTIN、Lot 及日期欄位之間沒有空格。

   d. 如果存在日期，便會以 YYMMDD 表示，而且以 0 填滿，不含 '/' 字元。

2. 使用標準 ANSI CRC-16 計算 PlainText ASCII 位元組的 ANSI CRC-16 雜湊 以 X16 + X15 + X2 + 1 的多項式雜湊

   請參閱 CRC16 函數。

3. 採用十進位格式 (雜湊模數 10000) 的最小 4 位數從雜湊計算 VoiceCode。十進位格式 (雜湊模數 10000)。

範例：

GTIN = (01) 10850510002011Lot = (10) 46587443HG234

PlainText = 10850510002011146587443HG234

CRC-16 Hash = 26359

VoiceCode = 6359

Large Digits = 59

Small Digits = 63

範例：

VoiceCode("10850510002011", "46587443HG234") = 6359

VoiceCode("65457886676767", "2", "100126") = 5836

ATA 2000 轉換函數

注意：Windows 系統區域設定可能會影響某些公式的輸出。

Bin2Hex(«value», «pad»)

將二進位值轉換為十六進位輸出值。 “pad” 參數用於通過在結果的左側添加 0 字元來定義整個輸出的長度 .

範例:

Bin2Hex("0010100111101001101",0) 將返回 14F4D

Bin2Hex("0010100111101001101",0) 將返回 00014F4D

CRC16_CCITT(«string»)

計算 CRC16 CCITT 校驗碼 .

在針對 TOC 的 ATA2000 規則中使用此函數來計算 .

範例:

CRC16_CCITT ("A") 將返回 B915

CRC16_CCITT ("AB") 將返回 4B74

CRC16_CCITT ("ABC") 將返回 F508

CRC32_CCITT(«string»)

計算 CRC32 CCITT 校驗碼 .

在針對 TOC 的 ATA2000 規則中使用此函數來計算 .

範例:

CRC32_CCITT ("A") 將返回 D3D99E8B

CRC32_CCITT ("AB") 將返回 30694C07

CRC32_CCITT ("ABC") 將返回 CBF43926

Dec2Bin(«value», «pad»)

將十進位值轉換為二進位輸出值。“pad”參數用於通過在結果的左側添加 0 字元來定義整個輸出的長度 .

範例:

Dec2Bin("5", 0) 將返回 101

Dec2Bin("5", 4) 將返回 0101

Dec2Hex(«value», «pad»)

將十進位值轉換為十六進位輸出值。“pad”參數用於通過在結果的左側添加 0 字元來定義整個輸出的長度 .

範例:

Dec2Hex(510, 0) 將返回 1FE

Dec2Hex(510, 4) 將返回 01FE

String2Hex(«value», «pad»)

將字串值轉換為十六進位輸出值。“pad”參數用於通過在結果的右側添加 0 字元來定義輸出的對齊方式（以位元組為單位）.

在 RFTAG 中，資料通常以使用詞對齊（2 個位元組）的十六進位進行編碼 .

範例:

String2Hex("A", 0) 將返回 41 (**A** 是十進位中的 AINSI 代碼 **65**，在十六進位中是 **41**)

String2Hex("ABC", 2) 將返回 414243

String2Hex("ABC", 4) 將返回 41424300

下面是 RFID ATA2000 出生記錄的樣本 :

String2Hex("MFR S0671*SEQ M37GXB92*PNO PQ7VZ4*PDT CONTROLLER*ICC 456789*UIC 2*DMF 20160701*UNT KG*HAZ UN0003*ESD 1*LOT 123456*CNT FR*PMLPML123456789", 2) 將返回

4D46522053303637312A534551204D333747584239322A504E4F20505137565A342A50445420434F4E54524F4C4C45522A49

String6BitsEncoding(«value», «pad»)

將字串值轉換為 6 位元 ASCII 編碼輸出值 .

“pad” 參數用於通過在結果的右側添加 0 字元來定義輸出的對齊方式（以位元組為單位）.

在 ATA2000 中，資料可以使用 6 位（而不是 8 位元）進行編碼以便壓縮 .

範例:

String6BitsEncoding("11", 0) 將返回 C710

VDAString6BitsEncoding(«字串 »)

將字串值轉換為 6 位元 ASCII 編碼 並採用德國汽車工業協會（VDA）的要求。 在 ATA2000 中，可以將資料編碼為 6 位元（而不是 8 位元）以進行比較。 用於轉換的表格可在**此處**找到。欲了解更多詳情，請參閱規格 VDA5500 和 VDA4994。

範例:

VDAString6BitsEncoding("11") 將返回 C71860820820

日期和時間函數

以下日期變數代表系統日期和時間，而非由程式定義的日期變數。

注意：Windows 系統區域設定可能會影響某些公式的輸出。

BestBefore(**date , dateFormat , offset , offsetUnit , changeMonth , outputFormat, localeID**)

可讓您根據鍵盤輸入值來計算有效期限的日期 (並非根據目前的日期)。

傳回值	字串	一種依照基準日期和位移所計算的字串格式日期值。
date	字串	字串格式的基準日期值
dateFormat	字串/ 數字	基準日期值的格式。 - 由數字編碼的預先定義格式。(例如，1-"dd/mm"、2-"dd mmmm" 等,詳細見下文表) . 可能的值可在這裡找到。 - 0 或負值表示使用目前的系統日期格式 - 由字串值直接編碼的格式。(例如 "yyyy/mm/dd")
offset	數字	要加入基準日期中的日期單位數。

offsetUnit	字串/ 數字	offset 參數的意義。 - 1、'' 或 '' 代表天數 - 2、'' 或 '' 代表月數 - 3、'' 或 '' 代表年數
changeMonth	數字	選擇性 (預設值：True) - 0 代表 False - 1 代表 True 表示計算的日期並不存在於該月中 (例如 2 月 30 日) True 會傳回下個月的第一天 (例如 3 月 1 日) False 會傳回當月的最後一天 (例如 2 月 28 日)
outputFormat	字串/ 數字	選擇性 (預設值：與 dateFormat 相同) 此參數會指示計算日期的輸出格式。可能的值會與 dateFormat 的情況相同。
localeID	數字	選用（預設值： 0, 會使用系統的地區設定） 此參數可指用用來顯示日期值的地區設定。

例如:

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1033) will return December 14, 2013.

BestBefore("14/06/2012" ,"dd/mm/yyyy","18","m", 1, "mmmm dd, yyyy", 1036) will return Décembre 14, 2013.

例如: Formula with a date data source.

Create a date variable name date0 with the date of the day: 26/06/2012 for this exemple with dd/mm/yy format.

BestBefore(date0 ,"dd/mm/yy","18","m", 1, "dd/mm/yyyy") will return 14/12/2013.

CFIA_Month(«date»):

傳回加拿大食品檢驗局（CFIA）使用的月份名稱：JA, FE, MR, AL, MA, JN, JL, AU, SE, OC, NO, DE。

例如:

CFIA_Month ("03/01/2021") 傳回 MR

CFIA_Month (today()) 根據今天的日期傳回月份名稱，如果今天是 01/01/2021 - 傳回值為 JA an>

CFIA_Month ("06/01") 傳回 JN

附註：「«日期»」參數格式取決於系統所應用的環境設定。

DateOffset(date , offset, offsetUnit , changeMonth)

傳回值	日期	將日期間隔加入指定的基準日期中。
date	日期	基準日期值。
offset	數字	加入基準日期中的日期單位數。
offsetUnit	字串/數字	offset 參數的意義。 - 1、'' 或 '' 代表天數 - 2、'' 或 '' 代表月數 - 3、'' 或 '' 代表年數
changeMonth	數字	選擇性 (預設值：True) - 0 代表 False - 1 代表 True 表示計算的日期並不存在於該月中 (例如 2 月 30 日) True 會傳回下個月的第一天 (例如 月 1 日) False 會傳回當月的最後一天 (例如 2 月 28 日)

例如:

DateOffset("26/06/2012",1,"d",1) will return 27/06/2012

-OR-

DateOffset(today(),1,"D", 0)

例如: Formula with a date data source.

Create a date variable name Date0 with the date of the day: 26/06/2012 for this exemple with dd/mm/yy format.You should ensure that date variable is in default system format (otherwise you can use DateValue() function to get the date value from the date variable).

DateOffset(Date0,1,"D", 0) will return 27/06/2012

DateValue(formattedDate , format)

傳回值	日期	傳回 formattedDate 參數的日期值。
formattedDate	字串	將轉換成日期值的文字格式日期。
format	字串	選擇性 (預設值：系統日期格式) 指定的直接格式字串。例如 "dd/mm/yy"

例如:

DateValue(22062012,"ddmmyyyy") will return 22/06/2012.

day (*date*) ：可求出 date 引數的日期。

FiscalDate(«fiscalStartDate», «outputFormat»)

讓你計算財政年度的開始日期（ yyyy.mm.dd）。

例子（如果當前的日期是 2009 年 11 月 24 號）：

FiscalDate("2009.01.01", 1) = 09

FiscalDate("2009.01.01", "yyyy") = 2009

FiscalDate("2009.01.01", 3) = 47

FiscalDate("2009.01.01", "day") = 328

hour (*date*)：可求出 date 引數的小時。

minute (*date*)：可求出 date 引數的分鐘。

month (*date*)：可求出 date 引數的月份。

now ()：可求出當前的日期和時間。

second ()：授予日期引數的秒數。

shiftcode(items)

是標籤上欄位的資料來源，必須根據目前時間變更。一個輪班表示一段已定義並命名的時間間隔。

傳回值	字串	傳回目前的 shift code 值。
項目	字串	以下為 shift code 項目的格式：開始時:開始分-停止時:停止分-數值|…| 開始時:開始分-停止時:停止分-數值

例如，可以依照以下內容定義輪班：

- 7:00 至 15:00 = 日班
- 15:00 至 23:00 = 晚班
- 23:00 至 7:00 = 夜班

注意：

1. 請使用 24 小時制的時鐘來定義時間值。0:00 為午夜；12:00 為中午。

2. 如果定義中有任何時間重疊，就會傳回第一項可接受的值。

3. 如果目前時間沒有合適的輪班項目，就會傳回空白字串。

範例（假設目前時間為 16:00）：

shiftcode("7:00-15:00-日班|15:00-23:00-晚班|23:00-7:00-夜班") ="晚班"

SpecificDateFormat("[date format]", "+/[offset data][date interval]")

SpecificDateFormat 函數允許您以特定間隔抵銷日期，以自訂化日期戳記並在方程式運算式使用此自訂化日期。SpecificDateFormat 函數會依據系統時鐘，使用預設日期格式或選擇的不同格式
，**抵銷目前日期。**

在運算式使用時，SpecificDateFormat 函數必須使用以下參數編寫：

SpecificDateFormat ("[date format]", "+/ [offset data][date interval]")

範例：SpecificDateFormat("mmmm","+2M")

- [date format] **指定要使用的日期格式。此參數必須包含在方括號內。**檢視有效日期格式***的表格***.

日期設定	輸出	描述
D 和 d	1	月份日期（沒有前置零）
DD	01	2 個字元長的月份日期（前置零）
dd	1	2 個字元長的月份日期（前置空白）
DDD	224	年初至今的天數（前置零）
ddd	224	年初至今的天數（前置空白）
DDDDD 和 ddddd	33482	自 1/1/1900 至今的天數 （可使用 5 到 9 個字元）
W 和 w	1	星期的特定天數（Sun=1，Sat=7）
WW	34	年初至今的週數

		（前置零）
ww	34	年初至今的週數 （前置空白）
WWW 和 www	783	自 1/1/1900 至今的週數（後 3 位數）
WWWW 和 wwww	4783	自 1/1/1900 至今的週數 （可使用 4 到 9 個字元）
WWWWWWWWW	000004783	自 1/1/1900 至今的週數（前置零）
wwwwwwwww	4783	自 1/1/1900 至今的週數（前置空白）
M 和 m	9	月份位數（沒有前置零）
MM	09	2 個字元位數月份（前置零）
mm	9	2 個字元位數月份（前置空白）
MMM	009	3 個字元位數月份（前置零）
mmm	9	3 個字元位數月份（前置空白）
MMMM 和 mmmm	1101	自 1/1/1900 至今的月份數
MMMMM	01101	自 1900 年至今的 月份數（5 個字元長）（前置零）
mmmmm	1101	自 1900 年至今的 月份數（5 個字元長）（前置空白） （可使用 5 到 9 個字元）
MMMMMMMMM	000001101	自 1900 年至今的 月份數（9 個字元長）（前置零）
mmmmmmmmm	1101	自 1900 年至今的 月份數（9 個字元長）（前置空白）
YY 和 yy	91	年份（2 位數）
Y 和 y	1991	完整年份
YYY 和 yyy	991	年份後 3 位數 （可使用 3 到 9 個字元）
YYYYYYYYY	000001991	完整年份（前置零）
yyyyyyyyy	1991	完整年份（前置空白）

- +/[offset data] **此值指定抵銷日期的數量。依據您想新增至目前日期或減去目前日期，間隔必須前置加號 (+) 或減號 (-)。**
- [date interval] **日期間隔必須為** d（**天數**）、w（**週數**）、m（**月份數**）**或** y（**年數**）。

以下表格顯示以不同格式在運算式使用 SpecificDateFormat 函數的範例。（注意：在這些範例中，假設目前時間為 2013 年 7 月 29 日）

SpecificDateFormat("mm dd yy" , "+1y") = 7 29 14 （**目前日期的** 1 **年後**）

SpecificDateFormat("ww/m/yyyy" , "30w) = 1/12/2012 （**目前日期的** 30 **週後**）

SpecificDateFormat("ddddd / wwww /mmmm" , "") = 41484/5928/1363 （**自** 1900 **年至今的天數、週數和月份數**）

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"time interval +/offset time")

TimeOffset 函數允許您以特定間隔抵銷時間，以自訂化時間戳記並在方程式運算式使用此自訂化時間。TimeOffset 函數會依據系統時鐘，使用預設時間格式，抵銷目前時間。

在運算式使用時，TimeOffset 函數必須使用以下參數編寫：

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"time interval +/ offset time")

範例：TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5")

- **您僅可使用「**mm/dd/yyyy hh:nn:ss**」的預設格式。要輸入不同格式，您需要同時包含** TimeOffset **和** DateValue **函數，然後使用可指定其他時間格式的** FormatDate **函數。**
- 「+/offset time」 **此值指定抵銷時間的數量。依據您想新增至目前時間或減去目前時間，抵銷必須前置加號 (+) 或減號 (-)。**
- 「time interval」 **時間間隔必須為** s（**秒**）、m（**分鐘**）**或** h（**小時**）。

以下表格顯示以不同格式在運算式使用 TimeOffset 函數的範例。（注意：在這些範例中，假設目前時間為 2013 年 7 月 30 日下午 4:12:10）

TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"),"h+5") = 07/30/2013 21:12:10 （**目前時間的** 5 **小時後**）

FormatDate(DateValue(TimeOffset(FormatDate(Now(), "mm/dd/yyyy hh:nn:ss"), "n+10"), "mm/dd/yyyy hh:nn:ss"), "hhnn") = 16-22 （**目前時間的** 10 **分鐘後**）

today ()：可求出當前的日期和時間。

Week *(date)* ：可求回 date 引數的週數。

weekday (*date*)：可求回 date 引數的星期。

注意：星期日被認定為一週的開始日。

例如：

on "Tuesday November 20, 1999" the weekday(now()) = 3

WeekISO8601(Date, DateFormat)

允許您建立支援 ISO8601 的公式。ISO 8601 是一種包含日期與時間相關資料交換的國際標準。

傳回值	日期	傳回指定日期的週數
Date	字串	要轉換成日期值的文字格式日期。
DateFormat	字串	基礎日期值的格式。 - 數字編碼預先設定的格式。 （例如 1 dd/mm/yy、2 dd/mm/yyyy 及其他 - 0 或負值表示使用目前的系統日期格式 - 字串值直接執行格式的編碼。（例如「yyyy/mm/dd」）

例如：

WeekISO8601(" 03/01/2010 ",0) = 53

WeekISO8601(" 02/01/2011 "," dd/mm/yyyy ") = 52

WeekISO8601(" 01/01/2011 ", " dd/mm/yyyy ") = 52

year (date)： 可求出日期引數的年份。

例如：

minute (now()) 可求出當下的時分。

year (today()) 可求出當日日期的年份。

邏輯函數

邏輯函數可讓您檢查是否符合一個或更多的條件。

注意：Windows 系統區域設定可能會影響某些公式的輸出。

注意：TRUE 等於 1 且 FALSE 等於 0

and (*expr_1*, *expr_2*) 若兩個引數皆為真，可求回 TRUE；若至少一個為偽，則求回 FALSE。引數必須從邏輯數值計算。

例如：

and(exact("string1","string"),exact("string","string")) = 0

and(exact("string","string"),exact("string","string")) = 1

exact (*string_1*, *string_2*) 的兩個字串相同，將求出 TRUE 數值；否則求出 FALSE 數值。此函數有大小寫之分。

例如：

exact("software","software") = 1

exact("software","software") = 0

if (*expr*, *Val_if_true*, *Val_if_false*) 如果 *Expr* 為真，可求回 *Val_if_true* 數值，如果 *Expr* 為偽，則求回 *Val_if_false* 引數。

例如：

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

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

not (*logical*) 可求出 *logical* 引數的相反數值。

例如：not(exact("string", "string")) = 1

not(exact("string", "string")) = 0

not(False) = 1 or not(O) = 1

not(True) = 0 or not(1) = 0

not(1+1=2) = 0

or (*expr_1*, *expr_2*) 若兩個引數之一為真， 可求回 TRUE；若兩個引數皆為偽， 則求回 FALSE。引數必須從邏輯數值計算。

例如：

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

數學函數

使用數值運算， 並得出數值性的結果。數值可以是變數或 常數。

注意：Windows 系統區域設定可能會影響某些公式的輸出。

Abs(data): 此功能函數顯示數據的絕對 (正) 值。允許在數字後使用字母。

例如：

Abs(-5) = 5

Abs(5) = 5

base10tobaseX(*string_1,string_2*) 可將 *string_2* 從 10 進數轉為 *string_1* 進數

例如：

如果一個名為 Base 16 的欄位含有 "0123456789ABCDEF"字串

BASE10TOBASEX(Base16, 12) 可求得 C

BASE10TOBASEX(Base16,10) 可求得 A

BASE10TOBASEX("012345","9") 可求得 13

注意：此公式不能接受負十進制數為 **" string_2 "**參數.

baseXtobase10(*string_1,string_2*) 可將 *string_2* 從 *string_1* 進數轉為 10 進數

例如：

如果一個名為 Base 16 的欄位含有 "0123456789ABCDEF"字串

BASEXTOBASE10(Base16, "E") 可求得 14

BASEXTOBASE10(Base16,10) 可求得 A

BASEXTOBASE10("012345","9") 可求得 13

注意：此公式不能接受負十進制數為 " val_1 "參數.

Ceil(data): 此功能函數會將數據四捨五入至下一個整數。允許在數字後使用字母。

例如：

Ceil(3.234) = 4

Ceil(7.328) = 8

Decimals(data1, data2): 此功能函數在 data1 使用 data2 小數位置。允許在數字後使用字母。

例如：

Decimals(4, 2) = 4.00

Decimals(3.524, 1) = 3.5

eval_add(«string»,«string»): 傳回參數的總和。

例如：

eval_add(5,5)=10

eval_div(«string»,«string»):傳回參數的相除值。

例如：

eval_div(20,2)=10

eval_mult(«string»,«string»): 傳回參數的相乘值。

例如：

eval_mult(5,2)=10

eval_sub(«string»,«string»): 傳回參數的相減值。

例如：

eval_sub(20,10)=10

Floor(data): 此功能函數會將數據四捨五入至下一個整數。允許在數字後使用字母。

例如：

Floor(3.234)= 3

Floor(7.328)= 7

hex(*val_1,val_2*) 可將 *val_1* 小數轉為十六進位格式，總數值為 *val_*

例如：

hez(2,8) = 00000002

int (*value*) 可求回低於或等於 *value* 引數的最大整數。

例如：

int (-5.863) = -6

int (5.863) = 5

max(data1, data2): 此功能函數顯示最小值。允許在數字後使用字母。

例如：

Max(5, 12) = 12

min(data1, data2): 此功能函數顯示最小值。允許在數字後使用字母。

例如：

Min(5, 12) = 5

mod (*val_1*, *val_2*) 可求回 *val_1* 引數除以 *val_2* 引數所得到的餘數。結果的正負號與除數相同。

例如：

mod (7,2) = 1

mod (-7,2) = 1

mod (7,-2) = -1

mod (-7,-2) = -1

quotient (*val_1*, *val_2*) 可求回 *val_1* 引數除以 *val_2* 引數所得到的整數結果。

例如：

quotient(10,2) = 5

round (*val_1*, *val_2*) 可求回進位至 *val_2* 所指定之位數的 *val_1* 引數。

- 如果 *val_2* 大於 0， *val_1* 會進位至指定的小數位數。
- 如果 *val_2* 等於 0， *val_1* 會進位至最接近的整數。
- 如果 *val_2* 小於 0， *val_1* 會進位至小數點以左的位數。

例如：

round (4.25,1) = 4.3

round (1.449, 1) = 1.4

round (42.6,-1) = 40

文字函數

注意：Windows 系統區域設定可能會影響某些公式的輸出。

如果表格內的每一欄包含一個字元，便可將一個字串類推插入表格內。此表格由其長度定義（字串的總字元數，包括空格在內）。字串內的字元 位置 分別對應其在表格內的位置，例如第一個字元就在第一個位置。例如：位置 3 對應至字串內的第三個字元。

cyclebasex（）可讓您在任何資料庫計數系統內執行計數。編號系統必須在連結之運算式內定義。每個號碼的目前值、每一次的遞增數值及重複數也必須預先指定。所有這些數值 都可連結至標籤內的其它欄位，但欄位名稱不得包含在引號內。

例如：

如有一個名為 Base 16 的欄位含有 0123456789ABCDEF 此字串，則：

cyclebasex(base16, "8", 1 ,1) = 8,9,A,B,C…

cyclebasex(base16, "F", -1,1) = F,E,D,C,B,A 9,8,7…

cyclebasex(base16, "B0 ", 1,1) = B0, B1, B2…

cyclebasex("012345", "4",1,2) = 4,4,5,5,10,10,11,11…

cyclechar（）可建立一個由用戶定義的完整循環的字元集。

例如：

cyclechar("A", "C") = A B C A B C A B C …

cyclechar("A", "C", 1,2) = A A B B C C A A B B …

cyclenumber（）可供您設定自己的號碼順序，而不使用正常的號碼或字母順序（0、1、2... 或 A、B、C...）

例如：

cyclenumber(1,3) 將按照下列順序產生標籤：1 2 3 1 2 3 1 2 3...

cyclenumber(1,3,1,2) 將按照下列順序產生標籤： 1 1 2 2 3 3 1 1 2 2 3 3 1 1...

cyclestring *( )* 可供您利用一個完整循環作為遞增欄位來建立一組文字或字元。整個字串必須包含在引號內（""），且每一個文字或一組字元必須以分號（;）分隔

例如：

cyclestring("Mon ; Tue ; Wed ; Thu ; Fri ; Sat ; Sun") = Mon Tue Wed Thu Fri Sat Sun

以下範例是針對使用 O 和 I 以外所有英文字母的標籤。

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*)) 的兩個字串相同，將求出 TRUE 數值；否則求出 FALSE 數值。

例如：

exact("software","software") = 1

exact("software","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*) 會求回 key 引數第一次出現在 string 引數內的位置。string 引數的搜尋將從 start argument (start >= 1) 求回的位置開始搜尋。如果找不到 key 引數，則函數將重設為零。該函數可區分小寫和大寫字母。

例如：

find("Peter McPeepert","P",1) = 1

find("Peter McPeepert","p",1) = 12

left (*string*, *num_char*) 可求回擷取自 string 引數的字串。此字串從其中一個 string 引數位置開始，並具有與 num_char 引數相等的長度。

例如：

left("Peter McPeepert",1) = P

left("Peter McPeepert ",5) = Peter

len (*string*) 可求得 string 引數的長度。空格也被當作字元計算。

例如：

len("Paris, New York") = 15

len("") = 0

len(" ") = 1

lower (string) 可將字串內的所有大寫字母轉成小寫字母。

例如：

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

LTrim(«string»): 此功能函數將會自動裁減左側數據的任何多餘空格或前置空格。

例如：

LTrim(" No."): No

mid (string, start, num_char) 可求回擷取自 string 引數的字串。此字串從對應 start 引數 (start >=1) 數值的位置開始，並具有與 num_char 引數相等的長度。

例如：

mid("Paris, New York",8,8) = New York

pad *()* 可將字元新增至欄位左邊，為整個輸入指定一個預設長度。任何字元都可選作填充字元使用。

例如：

如果一個名為 GREETING 的欄位顯示 HELLO 的數值，則：

pad(GREETING,8,0) = 000HELLO

pad(5,3,0) = 005

pad("Nine",6,"a") = aaNine

replace (string, start, num_char, new_string) 可求回轉換後的 string 引數。從 start 引數內定義的位置計算的字元數（相等於 num_char 引數）已被 new_string 引數取代。

例如：

replace("Paris, New York",8,8,"Singapore") = Paris, Singapore

replacestring(«string», «old_string», «new_string») 使用指定的«new_string» 在字符串«string» 中替換另外指定的所有«old_string»。

例如：

replacestring("ABC12DEF12", "12", "") = ABCDEF

rept (*string*, *num_char*) 求回的字串，其 string 引數的重複次數如 num_char 引數所定義。

例如：

rept("Ah Paris!",2) = Ah Paris!Ah Paris!

right (*string*, *num_char*) 可求得由 string 最末幾個字元組成的字串，其長度相等於 num_char 引數。

例如：

right("Purchase order",5) = order

RTrim («string»): 此功能函數將會自動裁減右側數據的任何多餘空格或前置空格。

例如：

RTrim("Part ") :Part

search (*string*, *key*, *start*) 可求得 key 引數第一次出現在 string 引數內的位置。搜尋將從 start argument (start >= 1) 定義的位置開始搜尋。如果找不到 key 引數，則函數將重設為零。

例如：

search("Purchase order","order",1) = 10

search("Purchase order","c",1) = 4

Sort(«列表», «分隔符號», «方向»): 傳回已排序的項目清單。排序是透過字串比較來完成的。此清單是一個字串參數，包含多個帶有分隔符號（例如逗號或分號）的項目。方向為可選整數，指定排序方向，0 = 升序，1 = 降序。如果未指定方向，則預設為升序。

例如：

Sort(“3;5;6;1”, “;”) = “1;3;5;6”

Sort("one;two;three", ";", 1) =“two;three;one”

StrAfter(«data»,«start after», «length»): 此功能函數會取用字元後指定開始位置前相同長度字元長的字串。

例如：

StrAfter("1234-5678", '-', 3)= 取破折號之後的 3 個字元 (567)

StrAfter("1234-5678", '-')= 取破折號之後的所有字元（5678)

StrBefore(«data»,«start before», «length»): 此功能函數會取用字元前指定開始位置前相同長度字元長的字串。

例如：

StrBefore("1234-5678", '-', 2)= 取破折號之前的 2 個字元（34）

StrBefore("1234-5678", '-')= 取破折號之前的所有字元 (1234)

SuppressBlankRows («string»): 將返回一個帶有跳過空行的字串。它可讓 您建立物件，在其中放置您想要的欄位，並禁止空 欄位。

範例:

SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})　（Var0、Var1 和 Var2 是變數，char(10) 是表示新行的"¥n"符號）)

Variables/Values	Formula	將顯示
Var0 = "Doris" Var1 = "Bull Run Ranch" Var2 = "Aurora"	*SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})*	Doris Bull Run Ranch Aurora
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	*SuppressBlankRows({Var0} & char(10) & {Var1} & char(10) & {Var2})*	Craig Chicago
Var0 = "Craig" Var1 = "" Var2 = "Chicago"	*{Var0} & char(10) & {Var1} & char(10) & {Var2}*	Craig Chicago

SuppressEmpty(«列表», «分隔符號»)：從項目清單中刪除空白條目。此清單是一個字串參數，包含多個帶有分隔符號（例如逗號或分號）的項目。

例如：

SuppressEmpty("3;;;1", ";") = "3;1"

trim(*string*) 可求回轉換後的 string 引數。在字串開端及末端出現的所有空格已被刪除。兩個文字之間的空格數減至一個。

例如：

trim(" Purchase order") = Purchase order

trimall(*string*) 可求回轉換後的 string 引數。所有出現的空格已被刪除。

例如：

trimall("Paris / New York / Rome") = Paris/NewYork/Rome

Unique(«列表», «分隔符號», «比較符號»): 從項目清單中刪除重複的條目。此清單是一個字串參數，包含多個帶有分隔符號（例如逗號或分號）的項目。可選比較符號指示比較是否區分大小寫，0 = 區分大小寫，1 = 忽略大小寫。如果未指定比較符號，則預設區分大小寫。

例如：

Unique(“3;5;3;1”, “;”) = “3;5;1”

Unique("one;ONE;two", ";", 1) = “one;two”

upper (*string*) 可求得轉換成大寫字母的 string 。

例如：

upper("Purchase order") = PURCHASE ORDER

ztrim *( )* 可在完全為數字的欄位內從左邊開始刪除所有零。

例如：

如果一個名為 WEIGHT 的欄位顯示 000200 的數值，則：

ztrim(weight) = 200

定義 Formula（公式）資料來源的屬性

指令：「**資料來源**」>「公式」>「內容...」

在**文件瀏覽器**的**資料來源**標籤內定義變數內容。

1. 直接在 **Edit（編輯）**方塊中，輸入公式。

   - 或 -

   選取想要選擇的元素，然後按一下 Insert（插入）。

2. 按一下 **OK（確定）**。

提示：連按兩下元素就可以插入該元素。

注意：如果公式中所用的變數名稱含有下列任一字元：&+-*/<>=^%,!¥"，該字元必須置於括弧 {} 中。

注意：動態預覽，表示包括「輸出」頁面中所定義格式在內的當前公式計算結果。 如果出現錯誤，預覽將顯示為紅色。 如果運算後的值被截斷，您必須在 Output（輸出）標籤中修改指定的最大長度。

實際操作 - 計算一個特定「模組 Modulo」

在此練習中我們將 EAN8 條形碼"Customer_Code"轉換為一個 2/5 Interleaved 碼，我們使用公式"Formula_4_NewCustCode"來完成這個轉換.

條形碼有如下的一些屬性:

- 符号体系: 打印机,
- 高度: 4 毫米,
- 窄条宽度: 1 毫米,
- 比率: 2,
- 人工识别符: 下面 / 居中,
- 与条形码之间的距离: 0 毫米,
- 字符字体: 打印机字体.

1. 打开标签文件 ORDER_WS2.LAB.

計算重量

創建一個公式並命名為 Formula_1_Weighted. 計算規則為：變量 Customer_Code 的第一個字符乘以 1, 第二個乘以 2, 第三個乘以 1, 第四個乘以 2, 等等.

變量的最大輸出長度為 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

將計算出的重量結果加起來:

接下來的一步是將前面公式中得到的結果加起來. 最大的輸入長度為 2.

創建第二個公式並命名為 Formula_2_Sum.

計算校驗位:

利用前面的結果， 我們來計算校驗位的值,

創建第三個公式並命名為 Formula_3_CheckDigit.

表達式如下:

if ((Formula_2_Sum % 10) > 0,10 - Formula_2_Sum % 10,0)

將原來條碼數值與校驗位組合起來:

當創建條形碼的時候必須包括原有數值和校驗位(Formula_3_CheckDigit).

創建第四個公式並且命名為 Formula_4_NewCustCode. 此公式將變量 Customer_Code 與校驗碼 Formula_3_CheckDigit 串接起來.

創建條形碼:

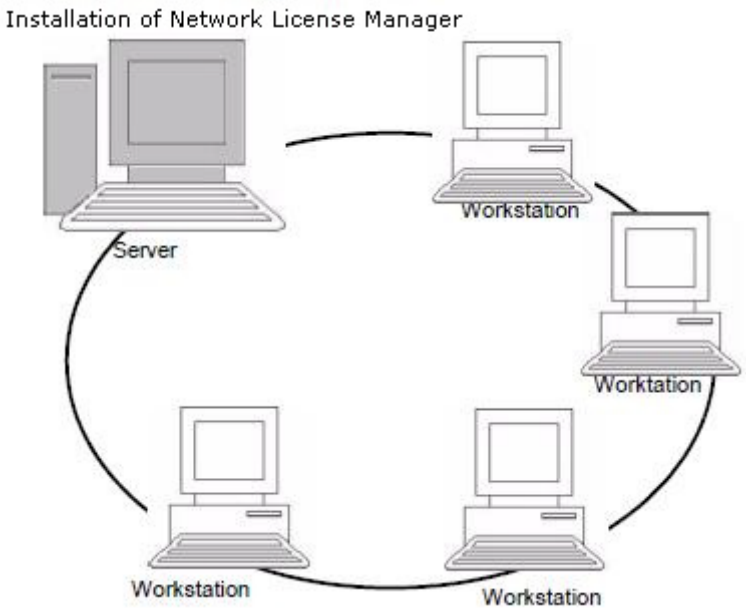
1. 選擇公式 Formula_4_NewCustCode 並將它拖到設計區域 Customer_Code 條形碼的位置.
2. 設置條形碼的屬性.

安裝 Network

功能說明

網路（多重使用者）套件可讓您透過網路，控制存取標 籤設定軟體的授權。 使用此公用程式，您可以讓許多使用者， 同時從網路上的任一位置，存取標籤設計軟體。

若要使用此標籤軟體的網路/ 多重使用者版，您必須在伺服 器或作為伺服器的工作站上安裝 Network License Manager， 然後在每個工作站上，安裝標籤設定軟體。



安裝程序

開始安裝前

Network Manager 必須先定義使用者群組的網路結構，您才可開始安裝軟體，具體說明如下：

- 定義安裝 **Network License Manager** 與硬體鎖 (Dongle) 的授權伺服器。
- 定義要使用標籤設定軟體的工作站或用戶端工作站

Network License Manager 說明

Network License Manager 可讓您使用標籤設定軟體的網路 組態設定。 Network Manager 包含：

- **Network License Manager (License Service)**
- **Network Settings Wizard： Network Settings Wizard 可協 助您定義網路組態設定。**

安裝 Network License Manager

在所有使用標籤設定軟體的工作站上安裝軟體之前，您必須 先在伺服器上安裝「**Network License Manager**」公用程式來設定網路 組態設定。

1. 將安裝光碟放入適當的光碟機中。

   接著會顯示安裝視窗

   如果未自動執行光碟片：

   前往 [Windows **檔案總管**] **並展開** DVD **光碟機。按兩下** a（例如：D:¥index.hta)。

2. 選擇 **Network License Manager**，該程式包括 **License Service**。 然後按一下安裝按鈕。
3. 依螢幕上的指令進行.
4. 如果您想要定義網路組態設定值，應啟動伺服器上的 **Network Settings Wizard**。 根據預設值，

   如果您未修改組態設定，每個工作站將使用各自的設定值。

組態設定

設定網路版需要的所有工具都可從 **Network License Manager toolbar**（可從 Windows 工作列 (Systray) 存取）取得。

從 Windows 工作列，按兩下圖示  以顯示 **Network toolbar**。

Network Settings Wizard 可幫助您定義網路版的設定。

1. 要啟動 **Network Settings Wizard**，請按一下圖示。
2. 在精靈中的步驟 1 中，選擇一個設定模式：一般、按照使用者或按照工作站。

    - **一般**：所有使用者在所有工作站上，會使用相同設定。(user.ini)
    - **按照使用者**：每位使用者可在任何工作站上存取其個人設定值。(user name.ini)
    - **按照工作站**：每個工作站都有專用的設定。(station.ini)

3. 在步驟 2 中，指定您要儲存這些設定值的位置。 如果您要在不同工作站共用這些設定值，請指定所有工作站都可存取的網路路徑。 （範例：TKDongle）。

4. 在步驟 3 中，指定您要儲存共用資料（變數、清單、列印日誌檔等）的位置。 請確定所有的使用者都有這些資料夾正確的存取權限。

啟動 License Service

在所有工作站上安裝標籤設定軟體之前，您必須先啟動 License Service。

會以服務的形式安裝 License Service。 您不需要啟動該程式。 事實上，會在工作站開機時啟動服務，並在工作站運作期間持續以背景作業方式執行。

啟動 Service Controller

- 按一下網**路工具列上的** 圖示，

  或

- 是連按兩下 TkxWebLicenseServerController.exe 檔案。
- 按右鍵 Windows 工作列中的**網路工具列**圖示，然後選擇**授權服務控制器**

在工作站上安裝軟體

所有使用標籤設定軟體的工作站都必須安裝此軟體。

在工作站上安裝軟體

1. 將安裝光碟放入適當的光碟機中。

   接著會顯示安裝視窗

   如果未自動執行光碟片： **前往** [Windows **檔案總管**] **並展開** DVD **光碟機。按兩下 a（例如：D:¥index.hta**)。

2. 選擇要安裝的產品，按一下安裝按鈕，並按照螢幕上顯示的指示進行安裝。

3. 啟動標籤設定軟體。 從工具功能表，選擇網路管理。

   - 或-

   從 Windows「開始」功能表，選取標籤軟體群組中的「網路管理」捷徑。

4. 啟用使用網路授權。

5. 選擇**網路授權類型**。

   - **共用資料夾授權**將使用 Windows 的檔案共用功能，在軟體和授權管理器之間進行通信。

   - **Web 授權**將在軟體和授權管理器之間使用 http/https 通訊。

6. 如果選擇 **Web 授權**類型，請指定**伺服器埠**。

7. 按一**下修改**以選擇要安裝 License manager 和硬體鎖的伺 服器。

   - 或-

   按一**下瀏覽**自動搜尋已安裝 License manager 的伺服器。

   如果已設定網路組態，會顯示是否要使用目前顯示的網路組態的訊息。

8. 如果您想要修改或設定網路設定，按一下 **Network Settings Wizard** 按鈕。

   按一下確定。

9. 點擊確定。

10. 重新啟動程式。



France
+33 (0) 562 601 080

Germany
+49 (0) 2103 2526 0

Singapore
+65 6908 0960

United States
+1 (414) 837 4800

Copyright 2026 TEKLYNX Corporation SAS. All rights reserved. LABEL MATRIX, LABELVIEW, CODESOFT, LABEL ARCHIVE, SENTINEL, PRINT MODULE, BACKTRACK, TEKLYNX CENTRAL, TEKLYNX, and Barcode Better are trademarks or registered trademarks of TEKLYNX Corporation SAS or its affiliated companies. All other brands and product names are trademarks and/or copyrights of their respective owners.

www.teklynx.com