

---

# LabelShop

## 条码标签打印软件

用户手册

Ver 1.2

北京科创京成条码科技有限公司

2002年10月



# 目录

第一章 概述 .....	1
系统简介 .....	2
系统运行环境 .....	4
软件的组成 .....	4
第二章 安装与卸载 .....	5
系统安装 .....	5
卸载软件 .....	7
第三章 了解标签格式 .....	9
概述 .....	9
标签设置 .....	11
标签格式中的对象 .....	17
第四章 创建及安排对象 .....	19
概述 .....	19
在标签格式中创建新对象 .....	19
安排标签中的对象 .....	21
关于对象的其它一些内容 .....	24

第五章 修改对象属性 .....	25
概述 .....	25
常规属性页 .....	25
字体属性页 .....	27
条码属性页 .....	28
文本属性页 .....	34
图形属性页 .....	36
图片属性页 .....	38
对象属性的附加说明 .....	41
第六章 对象的数据源 .....	43
概述 .....	43
常量 .....	45
序列号 .....	45
数据库 .....	46
日期 .....	47
时间 .....	47
键盘输入 .....	48
脚本 .....	48
其它高级选项 .....	57
字符数限制 .....	57
截短与验证 .....	58
子串列表与多子串操作 .....	59
使用数据工具 .....	62
第七章 从数据库中读取数据 .....	63
概述 .....	63

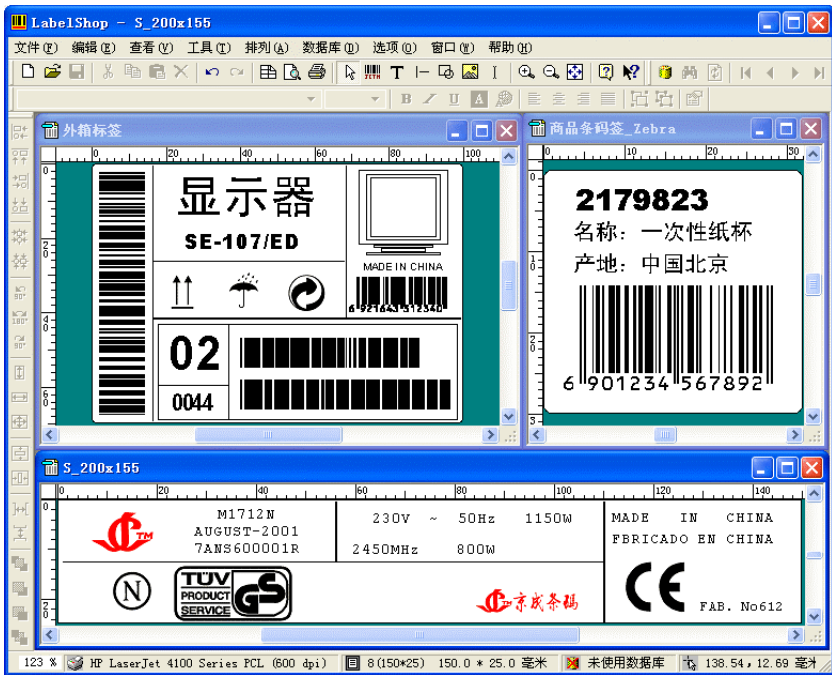
---

数据库类型 .....	63
数据库对话框 .....	65
ODBC 数据源连接示例 .....	67
第八章 打印输出标签格式 .....	73
概述 .....	73
打印对话框 .....	74
高级打印选项 .....	76
第九章 系统参数设置 .....	79
第十章 菜单命令参考 .....	81
附录 A 如何做 .....	87
附录 B 为什么 .....	91



## 第一章

## 概述



## 系统简介

LabelShop 是北京科创京成条码科技有限公司开发的条码标签打印软件。它的主要功能是进行条码标签设计输出，支持多种专业条码打印机和普通打印机，支持多种常用的一、二维条码，支持多种数据源格式。

### 支持多种打印机

LabelShop 不仅支持 Windows 下的所有打印机，还支持 ZEBRA、DATAMAX 等多种专业条码打印机。对于专业条码打印机，直接使用专用打印命令输出，充分发挥打印机的性能。

### 灵活多样的标签设计

LabelShop 的目的是让用户方便、快捷地设计制作出各种类型的条码标签。标签尺寸随意改变，页标签排列格式多种多样。标签中可以提供直线、矩形、条码、文字和图象等多种可视对象。

### 支持多种一维及二维条码

软件支持十多种码制的条形码，包括：

Code39	标准 39 码
CodaBar	库德巴码
ITF25	交叉 25 码
UPC-A	UPC-A 码（包括 2 位和 5 位附加码）
UPC-E	UPC-E 码（包括 2 位和 5 位附加码）
EAN-13	EAN-13 国际商品条码（包括 2 位和 5 位附加码）
EAN-8	EAN-8 国际商品条码（包括 2 位和 5 位附加码）
Martrix25	Martrix 25 码
Indestrial25	工业 25 码
China Post	中国邮政码
Code128	Code128 码 A、B、C 字符集，UCC/EAN128 码
PDF417	PDF417 码



## 丰富多样的字体

可以利用 Windows 下提供的所有英文和中文字体，对于条码打印机的内部专用字体，也提供了相应的支持。

## 多种数据源格式

用户可以为条码和文字对象的数据源定义多个变量，能够使用户方便灵活的为条码和文字对象定义数据，可以支持常量，序列号，数据库，日期，时间等多种数据类型，支持变量共享。一个对象的数据可由多个变量组成，一个变量可由多个对象共享。

## 多种图像格式

支持 bmp、gif、pcx、jpg、tif 等常用的图像格式。

## 数据库支持

LabelShop 提供读取 ODBC 数据库的功能，用户可以在标签中引用数据库中的内容。也可直接导入文本文件。

## 多种打印选项

可以打印输出页码和标签的对齐和裁切标记，标签的打印起点可任意设置，并且在连续打印时可跟踪标签位置。

## VBScript 脚本语言支持

LabelShop 支持 VBScript 语言，用户可使用脚本语言格式化数据，实现复杂的功能。

## 多语言支持

LabelShop 支持多种语言界面，不同语言可动态切换，并且可在同一标签中同时输出多种语言的文字。

## 全交互式操作界面

LabelShop 采用的是所见即所得的全交互式界面。用户可以使用键盘、鼠标来编辑、设计标签。同时，本软件提供了工具栏、状态栏、排列栏、属性栏、滚动条、标尺栏等操作工具。

## 系统运行环境

### 硬件环境：

CPU Intel Pentium 100 以上；

内存 32M 以上，建议 64M 或 128M；

硬盘必须至少有 50M 的可用空间；

CD-ROM 光盘驱动器(如果从 CD 安装系统)；

两键或三键鼠标；

显示器建议 800\*600 以上；

### 软件环境：

MS Windows 98

MS Windows 98SE

MS Windows Me

MS Windows NT 4、0 (Service Pack 3 以上)

MS Windows 2000

MS Windows XP

ODBC 数据库支持环境

## 软件的组成

LabelShop 软件系统由以下几部分组成：

CD-ROM 光盘

加密件（软件狗）

用户手册

---

**注意：**请用户在购买软件时，检查光盘上的序列号与软件狗上的序列号是否对应。如果存在任何问题，请与软件销售商联系并更换。

---

## 第二章

# 安装与卸载

## 系统安装

### 安装软件狗

关闭计算机主机电源及与主机连接的所有外围设备的电源；如果计算机主机后面的并行打印口接有打印信号线，将其拔下；将软件狗插在计算机后面的并行打印口上；将打印信号线接在软件狗的后面；重新打开与计算机连接的外围设备，并重新打开计算机主机电源。至此，软件狗安装完毕。

---

**请特别注意：**不要带电插拔软件狗，否则极有可能将其损坏。

---

### 安装软件

LabelShop 软件发行在一张 CD-ROM 光盘。其中包括安装程序、运行程序、帮助文件、样本文件、图象文件、字库文件及系统所需的系统支持环境。软件的安装从 CD-ROM 中的 Setup.exe 开始。

建议在安装 LabelShop 之前，关闭当前运行的所有应用软件。必要时可以重新启动计算机。

**步骤 1：**将 LabelShop 系统 CD-ROM 光盘放入 E：驱动器(假设 E：为光盘驱动器)。

**步骤 2：**在放入 LabelShop 光盘后，Windows 系统会自动运行光盘中的

安装程序。安装程序先弹出初始画面，之后，安装程序将收集有关信息，并生成一些用于安装的临时文件，同时调入安装软件所需的系统数据和模块。接下来显示的是“欢迎”对话框。其中有欢迎信息、操作信息及版权警告信息。用鼠标单击“下一步”按钮或按“回车”键，进入设置安装选项。如果想停止安装并退出安装程序，可用鼠标单击“取消”按钮。

**步骤 3:** 弹出“软件许可证协议”对话框后，这里显示本软件的许可协议，请仔细阅读后再作选择。文本框中没有显示出来的其它部分，可以通过按“PageDown”键，或用鼠标操作滚动栏来查阅。如果接受本许可证协议，可直接按“回车”键，或用鼠标单击“是”按钮。只有选择“是”按钮同意该协议内容，才能继续下一步的安装；如果选择“否”按钮，安装过程将终止。

**步骤 4:** 弹出“选择目标位置”对话框。这时在“目标文件夹”中显示的是系统默认的要安装软件的目录。如果采用系统默认的安装目录，用鼠标单击“下一步”按钮或按“回车”键，进入后面的操作。如果要更改安装软件的目录，用鼠标单击“浏览”按钮，进入“选择文件夹”对话框。“选择文件夹”对话框会列出所有可用的文件夹。可以从中选择一个已存在的文件夹，也能够输入一个新的文件夹名称。如果输入的文件夹名称不存在，则会显示一个消息框，询问是否要创建该文件夹。

**步骤 5:** 显示“设置类型”对话框。这里有典型安装（典型的）、最小安装（简洁的）和自定义安装（特定的）三种安装类型可供选择。其中：典型安装是向大多数用户推荐的一种安装类型，它安装最常用的选件。包括：程序文件、帮助文件、样本文件、字库文件、动态连接库及图象文件；最小安装是一种最节省磁盘空间和速度最快的安装类型，它只安装最基本的选件。包括：程序文件、帮助文件及动态连接库；自定义安装是建议有经验的用户使用的一种安装类型，它可以选择需要的安装选件。可供选择的包括：程序文件、帮助文件、样本文件、字库文件、动态连接库及图象文件；系统默认的安装类型是典型安装，如果直接用鼠标单击“下一步”按钮或按“回车”键，就可进行后面的操作。如果选择了自定义安装，再用鼠标单击“下一步”按钮或按“回车”键，就会出现“选择部件”对话框。这里列出所有可供选择的安装部件。有经验的用户可以灵活选择。用鼠标单击“下一步”按钮或按“回车”键，安装继续进行。

**步骤 6:** 显示“选择程序文件夹”对话框。其中显示出已经存在的文件夹和子文件夹（或者是程序管理器中的程序组）。可以从中选择一个已经存在的文件夹，或者输入一个新的程序组名称。如果指定的文件夹不存在，系统会自动创建。

**步骤 7:** 进入安装文件复制阶段。在此画面中，可以看到安装文件复制

的进度。在安装程序复制文件过程中，如果想终止安装过程，用鼠标单击“取消”按钮或按“回车”键。

**步骤 8:** 复制文件操作完成后，会出现“创建快捷方式”对话框。询问是否在桌面上创建快捷方式。如果选择“是”按钮，在程序安装完成后，将在桌面上生成一个快捷方式。这样，运行 LabelShop 的标签制作程序就十分方便。当然，如果不需要在桌面上创建快捷方式，就选择“否”按钮。

**步骤 9:** 最后出现“设置完成”对话框。其中有两项可供选择：重新启动计算机和稍候重新启动计算机。选择“重新启动计算机”后，再用鼠标单击“结束”按钮或按“回车”键，就退出安装程序并重新启动系统。选择“稍候重新启动计算机”后，再用鼠标单击“结束”按钮或按“回车”键，就退出安装程序但不重新启动系统。由于在安装过程中增加或更新了一些系统文件，要使这些更改产生作用，就必须重新启动计算机。否则，直接运行本软件可能会出现问題。

---

**特别提示:** 本程序运行时需要 ODBC 数据库支持环境，如果您的系统中没有安装 ODBC，请先安装 ODBC 环境!!

---

软件第一次运行时，需要输入注册码，您可在软件光盘包装上或手册中找到软件的注册码。注册码是和软件加密狗一一对应的。

## 卸载软件

当不需要 LabelShop 软件或需要重新安装 LabelShop 软件时，都需要进行软件的卸载操作。由于本软件提供了卸载程序，所以完成卸载操作比较容易。卸载本软件有两种方法：

### 方法一：通过软件提供的卸载命令

- ① 在 Windows98 中，用鼠标单击“开始”按钮→“程序”菜单 →“LabelShop” →“卸载标签制作软件”；
- ② 卸载程序弹出单击“确认文件删除”对话框。

如果选择“否”按钮，卸载程序将终止运行；

如果选择“是”按钮，则继续完成卸载操作。

在软件安装过程中，安装程序记录下了所有的事件用于卸载。比如创建文件、文件夹、程序、快捷方式及注册表项目等。当进行卸载时，

会删除安装时所创建的内容。

③ 卸载程序弹出“删除共享文件”对话框。由于安装软件时，需要一些共享的文件，这些文件除了本软件使用外，其它应用程序也可能使用。如果卸载本软件时删除了这些文件，其它使用这些共享文件的程序就可能无法运行。所以建议用户不要删除这些文件。用鼠标单击“所有均否”按钮，就可保留所有共享文件。

④ 在执行卸载程序过程中，会出现“从您的计算机上删除程序”对话框。在对话框中显示了一个进度条，它显示了卸载过程的进度。当卸载程序结束后，在对话框的底部会出现一条消息，提示用户卸载过程已经结束。当有的项目不能被删除时，该消息会提示用户卸载过程已经结束，而且有一些项目必须手工删除。用鼠标单击“明细”按钮，能够列出需要手工删除的文件列表。用鼠标单击“确定”按钮，可以结束卸载过程。

### 方法二：通过“控制面板”

① 在 Windows98 中，用鼠标单击“开始”按钮 → “设置”菜单 → “控制面板”；

② 进入“控制面板”窗口后，用鼠标双击“添加/删除程序”图标；

③ 进入“添加/删除程序 属性”对话框后，在“安装/卸载”标签中，用鼠标选中“LabelShop”项，然后单击“添加/删除”按钮；

④ 系统提示是否确认卸载 LabelShop 软件，用鼠标单击“是”按钮；以下的操作步骤与方法一的操作相同，请参考方法一的说明。

## 第三章

# 了解标签格式

## 概述

标签一般用来给物品作上标记，以便对它进行识别或分类。标签可以由多种材料制成，包括纸张、布料、塑料或金属。它可以永久地贴在物品的表面上，就象超级市场中打印在食品包装盒上的 EAN 条码，也可以象服装商店中衣服上的标签那样轻易地撕掉。

生成一个标签的过程很简单。只要使用鼠标、工具栏和菜单选项，就可以设计出带有条码、图片和文字的标签。我们把这种放置在标签查看区域中的对象集合叫作**标签格式**。同时，我们将标签的尺寸以及标签在页面上的排列方式等信息称为**标签设置**以示区分。

---

**提示:**如果创建了一个新标签，并不是只有在标签格式被保存后才能打印，只要出现在标签查看区域的内容都可以打印。不过，为了以后能够再次打印这个的标签，将标签格式保存起来是一个好的办法。

---

## 设计简单的标签

本应用软件中所有的菜单设置和对话框参数都是按照最常用的格式预先设置的。这些设置称作默认设置。

当创建一个新的标签格式，标签的外观会受默认设置的影响。所谓默认设置就是应用程序自动使用的选项或设置。您可以在新建标签时选择目标打印机和一个预先定义的标签设置，当然，这些设置可在标签编辑过

程中随意修改！

## 打印机默认设置

不管当前保存在标签格式中的打印机设置是什么，当创建一个新的标签格式时，您都可以手工选择任一打印机，打印机的设置仍采用默认设置。而当打开一个保存的标签格式时，系统会自动查找与此标签格式相对应的打印机，如果该打印机不存在，系统将打印机设置为当前默认的打印机。例如打开一个按照热转印式打印机设计的标签，而当前默认的打印机又是激光打印机，如果系统中有该热转印式打印机（安装了该打印机的驱动程序），由该标签格式仍在热转印式打印机上输出，如果热转印式打印机未安装（驱动程序），则输出会在默认的打印机上进行。在程序的状态栏中，您可找到当前标签对应的打印机。当创建一个新的标签时，一个选择标签格式对话框可供您选择一个用于打印输出的打印机和一个预定义的标签格式。软件支持的专用标签打印机的标签格式是按卷筒纸的格式定义的，而普通打印机的标签纸格式上按页式设计的。

## 使用 LabelShop 程序提供的标签设置

尽管创建标签设置的过程很简单，也很容易。然而，本应用程序还是提供了若干种常用的标签设置，它们对设计新的标签会很有帮助。用户可以根据需要定制自己需要的标签设置，用户定制的标签设置同样可保存并可重复使用。

## 创建新的标签格式

文件菜单中的新建命令用于创建一个新的标签格式文档。由于 LabelShop 允许同时打开几个标签格式文档，如果选择新建命令，而此时已存在的标签格式文档已经在标签查看区域中，LabelShop 将打开一个新的标签格式文档窗口，而不影响当前的窗口。可以使用窗口菜单中的打开窗口列表访问其它打开的标签格式文档。或用鼠标单击工具栏中的新建按钮。这也可以使用默认设置创建一个新的标签格式文档。

## LabelShop 支持的专用条码打印机

LabelShop 支持 Zebra、Datamax 等专用条码打印机使用打印机专用控制指令输出。专用条码打印机使用卷筒纸，而普通打印机使用页式的纸张。对于普通的打印机，软件采用图形方式进行输出，而对于支持的专用条码打印机，软件采用专用的条码打印机控制命令进行输出。采用命令方式的优势在于其相对于图形方式，需要发送给打印机的数据量很小，因此可以提高打印机的输出速度，充分发挥专业打印机的性能。但是由于不同品牌条码打印机的控制命令各不相同，因此能够实现的功能也有差异，如不同的打印机具有不同的内部专用字体等。在 LabelShop 中已经



考虑到了这些差异，因此在采用“命令方式”输出时，同一标签在不同的打印机上输出结果可能有些差异。可以采用“标准驱动”方式以图形方式输出出来避免这些差异，但这样可能会有一些性能上的损失，主要表现在输出速度上。

## 标签设置

标签设置对话框包含控制标签的尺寸和标签在页面中的排列方式的选项。(这些选项并不控制标签中条码和文字对象的排列，而只控制标签本身的尺寸和排列方式)

标签设置对话框还包含一个显示当前页面布局的预览窗口，对于普通打印机和系统支持的专用标签打印机，标签纸的示意显示略有不同，普通打印机是按页方式显示的，而专用标签打印机是按卷筒纸的方式显示的。由于要将很多的页选项都放入一个对话框中，标签设置对话框的下半部分包括通过选项卡控制的区域，与其它修改属性对话框中的属性页方式相同。标签设置对话框根据选择的五个标签（打印机、页面、标签、分块和其它），显示出不同的选项。

### 打印机属性页

用于设置打印机的有关信息。



**名称** 选项用于指定标签格式使用哪个打印机打印。系统允许从已安装的 Windows 打印机列表中选择一台打印机。默认的设置是最近一次在

Windows 控制面板打印机对话框中指定的默认打印机。

为方便起见，与打印机相关的设置都存储在标签格式文档中，包括纸张的大小和专用标签打印机的打印速度等设置。这样，下次在需要打印相同的标签时，要做的只是装入标签格式文档并打印，打印机参数仍按上次打印时的值进行设置。

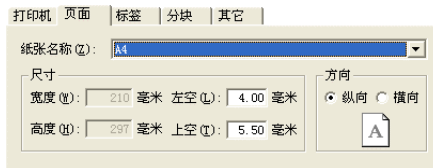
**重要说明：**在开始设计一个标签格式文档之前，首先需要选择一个合适的打印机，以便标签格式按照正确的打印机规格设置。这样做对于标签中包含条码对象的情况就格外重要。对于激光打印机和针式打印机来说，不同的条码密度范围可能会影响给定标签的尺寸。比如，当选择了一个激光打印机，放置在标签中的条码就可能比选择一个针式打印机时要短。

标签格式中可保存打印机的一些特殊选项，在下次打印输出时，会用同样的设置来配置打印机，而不论打印机的默认设置如何。同样，标签格式中也记住了对应的打印机，标签只在相应的打印机上输出，而不论该打印机是否是系统默认的打印机。

**驱动方式** 决定标签格式采用何种方式进行输出。对于普通打印机，只能采用标准的图形方式进行输出。对于系统支持的专用条码打印机，默认是按“控制命令”方式进行输出的，此种方式下，将直接使用打印机控制指令驱动打印机，能够充分发挥专用打印机的性能。您也可以采用“标准驱动”方式，以图形方式在专用打印机上进行输出。

**设置** 按钮弹出一个打印机设置对话框，可对打印机的参数进行配置。设置仅对当前标签起作用，不会改变打印机的默认参数设置。

## 页面属性页



页面属性页面用来设置整个页面的信息，除了“自定义纸张”外，其它纸张类型都有固定的尺寸。另外，对于卷筒纸，纸张的高度、上空和方向没有意义。

**纸张名称** 选项用于指定打印标签时所使用的纸张尺寸类型。许多打印机可以接受不同尺寸的纸张。普通的页式打印机最常用的纸张尺寸是

210×297 毫米，这种纸张尺寸叫做 A4。根据选择打印机的类型不同，会有其它尺寸的纸张可供选择。

当选择了某种打印机(特别是标签打印机)，尺寸选项列表框中可能包含“用户”或“用户自定义”的选项，以便自己指定纸张的尺寸。当选择了这个选项后，就可以输入标签介质的具体尺寸。

如果选择自定义纸张格式，请注意纸张的尺寸不要超出打印机的允许范围，否则可能产生预想不到的后果。

**宽度** 选项用于显示并设置纸张的宽度。但只有自定义纸张才能输入纸张的宽度。如果要允许使用这个选项，必须先从尺寸选项列表框中选择“用户自定义”项。

---

**重要说明：**当在热转印式打印机标签介质上打印，请记住在标签的衬底上通常能看出标签的边缘。因此，要指定的纸张宽度通常比页中标签的左边缘到右边缘之间的距离要大。可以采用以下步骤来补偿这种差异。

---

指定热转印式打印机标签介质的宽度：

1. 单击纸张名称选项列表框中“用户自定义”项。
2. 在宽度(纸张)选项中输入标签介质衬底的左右边缘之间的距离(不是标签的边缘)。
3. 在左空和上空选项中输入边空的数值。

**高度** 选项用于显示并设置纸张的高度。但只有自定义纸张才能输入纸张的高度。如果要允许使用这个选项，必须先从尺寸选项列表框中选择“用户自定义”项。

**方向** 选项用于设置打印机是以“纵向”还是“横向”打印标签。在出版业，“纵向”方向是传统的方式，页的高度大于宽度，而“横向”方向则是页的宽度大于高度。对于系统支持的专用标签打印机，此选项没有意义。

---

**重要说明：**虽然可以通过访问 Windows 控制面板中打印机设置对话框中打印机的方向选项更改页或标签的方向，但一般来讲，这样的设置对于标签中并不起作用。您需要 LabelShop 的“标签设置”对话框中设置这些参数。其它一些参数(如页面大小)也是一样。

---

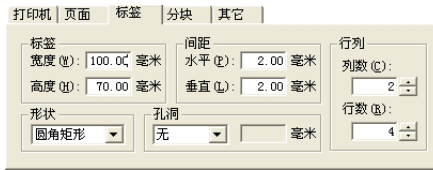
**上边空** 选项用于指定页的上边缘与第一行标签顶边之间的距离。

**左边空** 选项用于指定页的左边缘与第一列标签左边之间的距离。

通过调整页面的上边空和左边空，可以调整标签在页面中的相对位置。虽然调整这些数值也可以使得标签的实际打印位置与模切的标签对应起来，但一般建议将这两个数值设置成与标签纸的实际边空相同，而通过调整“其它”属性页中的位置微调来调整标签在页面中的位置，从而使实际输出的标签与模切的标签对齐。

**注意：**大多数激光和其它“页式”打印机无法打印到页的周边上一个很窄的区域中。这个“无法打印区域”会根据各种不同的打印机而有所变化。例如，HP 兼容激光打印机无法打印页边缘内大约 5 毫米的区域。因此，只将边空设置为零，并不意味着能够打印到纸张的边缘。

## 标签属性页



**标签宽度** 选项用于指定标签的宽度。标签的宽度是指从标签的左边缘到右边缘的距离。

**标签高度** 选项用于指定标签的高度。标签的高度是指从标签的顶边缘到底边缘的距离。

**水平间距** 选项用于指定标签的列间距。标签的水平间距是指从一个标签的右边缘到它右边一个标签左边缘的距离。

**垂直间距** 选项用于指定标签的行间距。标签的垂直间距是指从一个标签的底边缘到它下边一个标签顶边缘的距离。

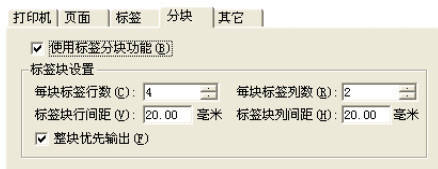
**列数** 选项用于指定标签介质上标签的列数。根据不同的标签介质，可能会有一列或多列的标签。

**行数** 选项用于指定标签介质上标签的行数。根据不同的标签介质，可能有一行或多行的标签。

**形状** 选项用于指定标签的外观形状。有直角矩形、圆角矩形和圆形三种外观选择。如果选择了圆形标签，则宽度和高度选项表示标签两个方向的直径，当标签高度与宽度数值相同时，则标签是一个正圆形标签。

**孔洞** 选项用于指定标签当中是否有洞。如果选择了圆洞，则有一个输入框可以输入孔洞的尺寸。孔洞位于标签的中心。

## 分块属性页



**使用标签分块功能** 决定是否将页内的标签分成块。标签分块功能可将一页内的标签分成几个块，标签块的间距可以与标签间距不同。

**每块标签行数** 设置一个标签块由几行标签组成。

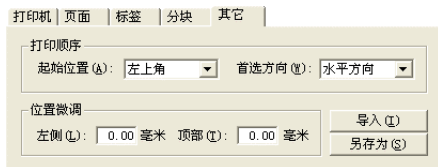
**每块标签列数** 设置一个标签块由几列标签组成。

**标签块行间距** 设置标签块间水平间距。

**标签块列间距** 设置标签块间垂直间距。

**整块优先输出** 设置当使用分块功能时，标签的排列先在一个整块内进行，当一个整块内排满后，才顺序排入下一个标签块。标签块的排列顺序和标签块内标签的排列顺序都取决于标签的启起位置和首选方向。

## 其它属性页



**起始位置** 决定在打印输出标签时，从整页的什么位置开始输出。标签通常从页面的左上角开始打印。然而，有时会需要改变输出顺序。标签设置对话框中的起始位置选项用于选择首先打印哪个标签。可以从以下四个设置中任选一个。

**左上角** 从左上角的位置(默认)开始打印。

**右上角** 从右上角的位置开始打印。

**左下角** 从左下角的位置开始打印。

**右下角** 从右下角的位置开始打印。

**首选方向** 决定标签按行或按列进行打印。标签的打印顺序默认是一行接一行打印。然而，有时希望标签的打印是按列进行。在这种情况下，就需要选择打印的首选方向。标签设置对话框中的首选方向选项用于选择打印顺序。可以从以下两个设置中任选一个。

**水平方向** 一行接一行打印。

**垂直方向** 一列接一列打印。

**位置微调** 选项用于指定标签打印的位置偏移。

**左侧偏移** 选项用于指定页的水平打印偏移。

**顶部偏移** 选项用于指定页的垂直打印偏移。

---

**说明：**一般来讲，页面的左空与上空同样可以移动整页的打印位置。但是用位置微调选项调整会更加直观方便。对于专用的条码打印机，可能只能往一定的方向进行微调。例如 ZEBRA 的条码打印机，只能向右或向下进行调整。因此，位置微调所产生的效果会因打印机的不同而不同。

---

**导入** 按钮用于调入预定义的标签设置。您可以在任何时候调入一个预先保存的标签设置。

**另存为** 按钮用于保存用户自定义的标签设置。用户自定义的标签设置被保存在“用户定义标签设置”中，您可以在任何时候调入一个系统预定义的标签设置一样重新调入一个用户保存的标签设置。

## 保存用户标签设置

LabelShop 可以方便地将用户自定义的标签设置保存起来，供用户象调用系统中定义的标签设置一样随时重复调用。

在其它属性页中，**导入**按钮用于导入一个程序定义的或用户保存的标签设置，**另存为**按钮将用户的标签设置命名保存。

## 关于标签模板

LabelShop 可以将用户标签格式保存成模板，以便重复使用。标签模板与标签格式的区别在于在标签模板中只能修改对象的数据，而无法修改对象的其它信息。

可以通过“另存为模板”菜单将一个标签格式保存为标签模板。可以通过系统选项中的设置决定是否在新建标签格式时弹出一个供选择标签模板的对话框。另外，打开标签模板时，系统的一些默认设置可能与打开普通标签格式时略有不同。

如果想对标签模板中的对象属性进行修改，必须将其另存为普通的标签格式才能进行。

### 标签格式附加说明

LabelShop 可以在每个标签格式中保存一些附加的说明信息，可以通过“文件——说明”菜单调出“标签格式附加说明”对话框进行设置。

对话框中的“下次打开时显示附加说明”设置用于设定下次打开标签时，是否自动弹出“标签格式附加说明”对话框。

## 标签格式中的对象

标签格式中可以排入文字、条码、线条、图形、图片对象。对象可以通过选中相应的对象工具，使用鼠标在标签上单击左键并拖动生成。对象的属性设置请参考后面章节。





## 第四章

# 创建及安排对象

## 概述

有两种方法在标签中添加条码、文字或各种图形对象。从工具菜单中选择适当的命令，或从工具栏中选择相应的按钮。在这两种方式中，使用任意一种的结果都是相同的。执行该命令后，鼠标指针会呈现出选定对象的形状，直到单击了其它按钮或菜单选项。当鼠标指针呈现出选定对象的形状时，通过将光标放置在希望的位置并**单击拖动**鼠标左键，就可以在标签查看区域添加一个该种对象。

在标签中创建对象后，对象的属性是系统默认的值。可以通过修改对象属性对话框中的选项来调整该对象的外观。

## 在标签格式中创建新对象

### 创建条码对象

1. 单击工具栏上的创建条码按钮。
2. 将光标移动到标签查看区域。当光标指针通过标尺进入标签查看区域时，会变为与工具栏中按钮相匹配的形状。这时就可以创建条码对象了。
3. 将光标移动到标签查看区域中需要创建条码框的一个角的位置，**单击鼠标左键并拖拽光标**，释放鼠标左键就可将条码对象放置到标签上。

## 创建文字对象

1. 单击工具栏中的创建文字按钮。
2. 将光标移动到标签查看区域。当光标指针通过标尺进入标签查看区域时,会变为与工具栏中按钮相匹配的形状。这时就可以创建文字对象了。
3. 将光标移动到标签查看区域中需要创建文字框的一个角的位置。**单击鼠标左键并拖拽光标**。释放鼠标左键。就可将文字对象放置到标签上。

## 创建线条对象

1. 单击工具栏中的创建线条按钮。鼠标指针会呈现线条光标的形状。
2. 将光标移动到标签查看区域中需要创建线条的开始位置。
3. **单击鼠标左键并拖拽光标**, 使出现的线条达到需要的尺寸。
4. 释放鼠标按钮。

只能创建水平或垂直的线条, 在创建线条时, 通过向不同的方向拖动鼠标指针, 可以创建水平或垂直的线条, 以后可通过旋转线条来调整线条的方向。

## 创建矩形对象

1. 单击工具栏上的创建矩形 按钮。鼠标指针会呈现矩形光标的形状。
2. 将光标移动到标签查看区域中需要创建矩形框的一个角的位置。
3. **单击鼠标左键并拖拽光标**, 使出现的矩形达到需要的尺寸。
4. 释放鼠标按钮。

## 创建图片对象

1. 单击工具栏中的创建图片按钮。
2. 将光标移动到标签查看区域。当光标指针通过标尺进入标签查看区域时,会变为与工具栏中按钮相匹配的形状。这时就可以创建图片对象了。
3. 将光标移动到标签查看区域中需要创建文字框的一个角的位置。**单击鼠标左键并拖拽光标**。释放鼠标左键。就可将默认的图片对象放置到标签上。
4. 双击图片对象, 在“图片” 属性页中选中“链接式图片”类型, 单击“浏览图片”按钮。

5. 在浏览图片对话框中，文件类型选项的默认设置是“所有支持的图象文件”，使应用程序能够显示出当前指定目录下的所有可用图片文件。要从指定图片文件格式列表中查看或选择需要的文件，只要从列表框中选择需要的文件格式即可。选中“预览图片”，可以查看供选择的图片。
6. 选中需要的图片，单击确定按钮，就可以在标签中看到选择的图片。

## 安排标签中的对象

排列菜单命令选项具有两方面的功能：

将多个对象结合成一组，组中的对象可以同时移动。

提供一些方法，用来更改对象彼此间的位置或尺寸。

通常使用基本的鼠标拖动就可以安排和移动标签中的对象。然而有时使用排列菜单中提供的功能来安排标签中对象的布局会更加方便。

如果想精确定位一个对象，可以通过修改对象的常规属性中的位置属性来实现。

### 选取一组对象

工具菜单中的选取命令用激活选取工具，并使用标准的鼠标光标来选择、移动和更改标签查看区域中对象的选项、位置和尺寸。选择一个对象操作就激活了用于屏幕上操纵对象的句柄，通过拖动对象本身或对象的句柄可以调整对象的尺寸和位置。也可以通过修改对象属性对话框中的选项来修改该对象的属性。

要激活选取工具并选择一个对象，使用下列一种选项：

单击工具栏中的按钮，然后再单击要选择的对象。或者单击工具菜单中的选取命令，然后再单击要选择的对象。

### 移到最前/移到最后/前移/后移

排列菜单中的移到最前和移到最后命令用于将选定对象移动到叠放在标签中其它对象的最前面或最后面。

---

**注意：**通过标签中重叠在一起的不同对象，可以创建出特殊图形和其它特别效果的标签。然而相互重叠的对象会使标签编辑变得复杂，这是由于选择最顶部的对象总是比底部的对象容易。对象还需要放置在合适的层次位置，才能达到期望的打印效果。另外，并不是所有的打印机都可以正确输出这些复杂的效果！

---

## 将多个对象组合与取消组合

排列菜单中的组合命令用于将标签格式文档中两个或多个对象结合成一个独立的组。在各个独立的对象结合成一组后，只要单击组中任意对象就可以选中整个组。一组对象具有与组中其它对象共有的属性还可以同时修改。例如，可以使用鼠标移动整组对象而不改变组内各个对象之间的相对位置。

要将对象结合成一个单独的组：

1. 使用下列方法之一就可以选择要结合的对象：按住 **SHIFT** 键的同时，在要选择的对象上单击，或者使用选取工具选择要结合成组的对象。
2. 选择组合命令。

---

**注意：**被组合起来的对象的属性是一般不可以直接更改的，如果要修改组合对象中的对象的属性，应先取消组合。

---

排列菜单中的取消组合命令用于将组合在一起的对象拆开。在对象取消组合后，只能同时选择拆开后的各个对象。

要将一组拆开成各个对象：

1. 通过单击组合对象中的一个，就可以选择要拆开的组。
2. 单击取消组合命令。

## 旋转对象

排列菜单中的旋转命令用于旋转标签格式文档中各个对象和对象组。旋转命令可以将对象不同的方式进行旋转：

**左旋 90°** 将选定的对象或对象组向左旋转 90°。

**旋转 180°** 将选定的对象或对象组旋转 180°。

**右旋 90°** 将选定的对象或对象组向右旋转 90°。

---

**注意：**旋转矩形框会自动改变它原来在图形属性对话框中设置的高度或宽度。旋转操作的参考点是对象的中心点，当多个对象被同时旋转时，将以所有被选取对象组合后的中心点为参考点进行旋转。

---

## 对齐一组对象

排列菜单中的对齐命令用于将标签格式文档中选择的多个对象按一定

的行列规则移动。这是通过将一组选定对象的水平和垂直排列设置为相同的属性来完成的。

多数对齐选项是用于排列两个或多个标签对象彼此之间的位置。因此，除非在标签中选择了两个或多个对象，否则这些选项多数是不可用的（灰色）。

参考对象是对齐操作的基准，参考对象总是很容易被识别，它有蓝色的实心句柄，而同一组中的其它对象只有黑色的实心句柄。一般第一个被选取的对象是参考对象。在熟悉对齐命令后，会发现位于标签查看区域左边的对齐栏中的快捷按钮很容易使用。

**左齐** 将选定的两个或多个对象的左边对齐参考对象的左边缘。

**水平中齐** 将选定的两个或多个对象的水平中间点坐标对齐参考对象的水平中线。

**右齐** 将选定的两个或多个对象的右边对齐参考对象的右边缘。

**顶齐** 将选定的两个或多个对象的顶边对齐参考对象的顶边缘。

**垂直中齐** 将选定的两个或多个对象的垂直中间点坐标对齐参考对象的垂直中线。

**底齐** 将选定的两个或多个对象的底边对齐参考对象的底边缘。

**水平居中** 将所选中的对象当做一个整体，放置在标签的水平中心位置。

**垂直居中** 将所选中的对象当做一个整体，放置在标签的垂直中心位置。

所有的对齐命令对于具有位置锁定属性的对象不起作用。

## 更改一组对象的尺寸

排列菜单中的尺寸命令用于将标签格式文档中选择的多个对象按一定的尺寸规则改变大小。这是通过将一组选定对象的水平和垂直尺寸设置为相同的属性来完成的。

多数尺寸选项是用于更改两个或多个标签对象彼此之间的尺寸关系。因此，除非在标签中选择了两个或多个对象，否则这些选项多数是不可用的（灰色）。

与对齐命令相同，所有的操作都是以参考对象为基准的。同时，一些对象的尺寸（如文字对象）只能通过更改属性来调整。

**水平同宽** 将选定的两个或多个对象的水平尺寸改变为参考对象的水平尺寸。

**垂直同宽** 将选定的两个或多个对象的垂直尺寸改变为参考对象的垂直尺寸。

**水平垂直相同** 将选定的两个或多个对象的水平和垂直尺寸改变为参考对象的尺寸。

## 均匀排布一组对象

排列菜单中的间距命令用于将标签格式中选取的多个对象均匀排列。

**水平间距相同** 这个命令与对齐命令不同,对齐命令需要选定两个或多个对象,而这个命令至少需要选定三个对象。系统可以分析选定对象相对于其它对象的位置,并且最靠近标签左边和右边的两个对象位置不变。在这两个对象之间的所有其它对象位置,则按照从左到右的顺序均匀地排列开。

**垂直间距相同** 这个命令与对齐命令不同,对齐命令需要选定两个或多个对象,而这个命令至少需要选定三个对象。系统可以分析选定对象相对于其它对象的位置,并且最靠近标签顶边和底边的两个对象位置不变。在这两个对象之间的所有其它对象位置,则按照从上到下的顺序均匀地排列开。

## 关于对象的其它一些内容

### 关于对象说明

可以为每个对象设置一些附加的说明信息,当设置了显示菜单中的显示对象说明项目,被选中的对象的说明信息的第一行将被显示出来。

## 第五章

# 修改对象属性

## 概述

在选定的对象上双击鼠标左键或选中属性菜单命令，会出现属性对话框，通过修改对象属性对话框中的选项，可以调整标签中对象的外观，以适应具体的应用环境。修改对象属性对话框也可以用来指定条码和文字对象的数据源，真正设计出用户所需的标签格式。

同时只能有一个对象的属性被修改，即便是同时选取了多个对象，修改的也只是参考对象的属性。

对象除了拥有位置、旋转等共同的属性外，不同类型的对象还拥有特殊的属性，如条码的高度，线条的宽度等

数据源属性描述了条码和文字以及数据源图片对象的数据内容及构成方式，数据源属性将在下一章进行专门讨论。

---

**注意：**不同的打印机对于对象属性的支持也不同，在特定的打印机上，一些属性可能无法实现。

---

## 常规属性页

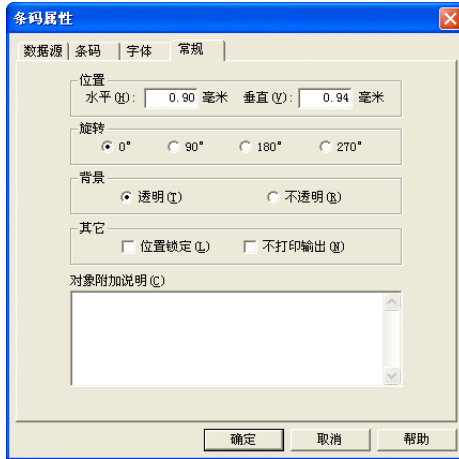
常规属性页中的选项用于指定放置在标签中对象的位置、是否打印、旋转角度、是否透明和对对象附加说明。

**水平位置** 用于指定标签对象的水平位置。即标签左边缘到对象左边缘的距离。

**垂直位置** 用于指定标签对象的垂直位置。即标签上边缘到对象上边缘的距离。

**旋转角度** 用于更改标签对象的旋转角度。默认的设置为 0 度，可以选择 90、180、或 270 度。

**透明方式** 用于指定标签对象与背景是否透明。



**位置锁定** 用于指定标签对象位置是否锁定。在编辑标签格式时，对象的位置确定后，为了在编辑其它对象时不影响这个对象的位置，需要将这个对象的位置固定住。锁定选项用于指定是否将该对象的位置锁定。

一旦选中对象的位置锁定选项，无论使用鼠标还是常规属性页，都无法移动该对象。使用鼠标时，鼠标的光标会变为“锁”的形状，而对对象不发生移动；使用常规属性页时，位置选项被禁止，无法更改其数值。

如果一个锁定的对象被组合，对象将失去位置锁定属性。

**不打印输出** 用于指定对象是否需要在打印机上输出。如果选中了该选项，则此在打印机上输出时将不包括此对象。这对于在标签上加入一些附加的说明是必要的。可能通过更改系统设置中的一些选项来输出非打印对象，也可以设置在编辑操作时不包括不些对象（这些对象不能被选取和编辑，这在需要一个标签参考背景时需要这样设置）。



**对象附加说明** 用于指定标签对象的附加说明。如果系统设置了“显示对象说明”（在查看菜单中），附加说明的第一行文字将在对象上显示。

可以使用此页中的位置设置可以精确的设置对象的位置。

## 字体属性页

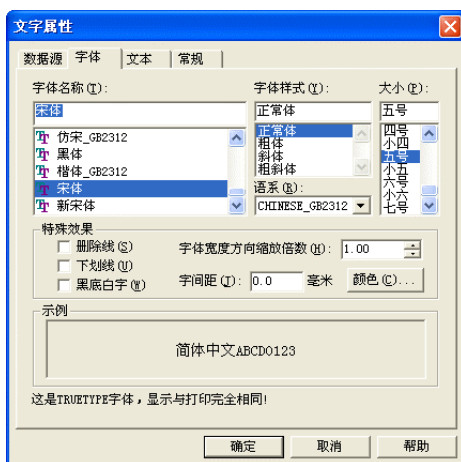
文字对象及含有可读性文字的条码对象拥有字体属性。

字体属性页中的选项用于设置标签中需要输出的文本的字体信息。**LabelShop** 可以使用由 Windows 提供的所有 TrueType 字体，对于支持的专用条码打印机，还包括打印机的专用字体。

TrueType 字体可以自由更改尺寸，也就是说可以指定需要的任何尺寸。它在屏幕上显示的尺寸总与打印输出的尺寸完全相同。使用 TrueType 字体，当选择新的打印机后，不必重新选择或安装新的字体。它可以在打印机之间自由更换，甚至在点阵打印机与激光打印机之间也可以自由切换，而不用调整字体类型。

对于软件支持的专用条码打印机，可使用打印机专用字体，专用字体的尺寸依赖于具体的打印机，一般不能连续调整。并且一些特殊选项也不一定支持。打印机字体的显示与输出效果不一定相同。

在修改条码属性和修改文字属性标签对话框中可以显示字体属性页。



**字体名称** 用于指定每个带有文字对象所使用的字体名称。

LabelShop 包含两种特殊字体，OCR-A 和 OCR-B，经常用于显示条码供人识读字符。由 Windows 提供，而且在 LabelShop 中可以引用的字体包括，Courier New、Symbol、Times New Roman，还有，宋体、黑体、楷体 and 仿宋体等，一些专用条码打印机还有一些专用字体。

**字体样式** 用于指定标签中文字是否采用正常体、粗体、斜体和粗斜体。

**大小** 用于指定标签中文字的大小尺寸。字体选项包含带有滚动条的列表框，以便快速更改文字的字体尺寸。对于打印机专用字体，字体的尺寸不能随意定义，只能是一些打印机可以支持的数值。

**特殊效果选项** 定义一些字体的特殊效果，并不是所有的打印机和所有的字体都可以支持这些效果。

**删除线** 用于指定文字是否带有删除线。

**下划线** 用于指定文字是否带有下划线。

**黑底白字** 用于指定文字是否是黑底白字，当选定此项时，颜色选项将失效。

**字体宽度缩放倍数** 字体宽度缩放倍数选项用于指定字体横向尺寸缩小或放大的倍数。对于所有文字对象的默认放大倍数是 1.00。然而，这个数值可以按照需要增加或减少。大于 1.00 的数值放大文本字体，而小于 1.00 的数值缩小文本字体。例如，给定文本对象字体宽度缩放倍数为 1.75，应用程序将文字放大到正常尺寸的 175% ( $1.75 \times 100$ )。同样，给定文本对象字体宽度缩放倍数为 0.45，应用程序将文字缩小到正常尺寸的 45% ( $0.45 \times 100$ )。

---

**重要提示：**当使用字体宽度缩放倍数选项放大文字对象时，需要注意以下几点：在本应用程序中，TrueType 字体可以缩放。然而，打印机字体就不能随意缩放，只能以打印机允许的数值进行缩放。当指定字体宽度缩放倍数时，输入的是只是近似值。软件将给出最接近的放大倍数。

---

**颜色** 用于指定文字对象和条码供人识读字符使用的颜色。

**字间距** 用于调整字与字之间的距离。需要注意的是，一些打印机专用字体的间距是无法进行调整的。

## 条码属性页

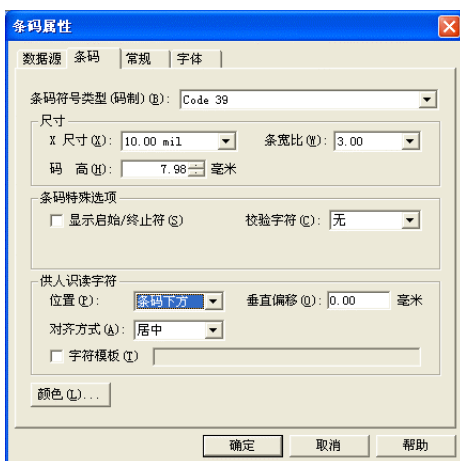
条码属性页中的选项用于指定标签中生成条码对象的外观。显示条码属性页最容易的方式是用鼠标双击要修改的条码对象。

**注意：**条码属性页中包含动态区域，可以根据选择条码的符号类型显示不同的选项。例如，如果条码符号类型设置为“ITF25”，动态区域中就会显示校验字符选项。然而，如果条码符号类型设置为“Code 128”，动态区域中就会显示 UCC/EAN-128 和字符集选项。如果选择了一组包含不同符号类型的条码对象，动态区域只显示当前主选定对象的选项。

条码选项列出如下：

**符号类型** 用于指定条码的符号类型(条码码制)。

**颜色** 用于指定标签内条码对象中条的颜色。



**X 尺寸** 用于通过修改窄条的“X 尺寸”指定条码的密度。(适用于所有条码符号类型。)

条码属性页中的 X 尺寸选项用于指定条码的最基本单元——最窄单元的宽度。这个最窄单元的宽度就称作条码的 X 尺寸。连同条码的宽窄单元比例，X 尺寸是决定条码密度的因素之一。减少条码 X 尺寸的数量就可以增加密度，同时增加条码 X 尺寸的数量就可以减少密度。

假设打印一个 Code 39 码(条宽比选项设为“自动”)到一个 600 dpi 的 HP 激光打印机上。可以指定的最小 X 尺寸数值为 1.67 mil。另一方面，最高的 X 尺寸数值为 50.0 mil。

**注意：**X 尺寸依赖打印机的精度，因此对于不同的精度的打印机，可供选择的数值也不相同，并且数值不能连续调整。

**条宽比** 用于指定条码中宽单元和窄单元的比例。(适用于 Code39、ITF25、Codabar,和 PDF417。)

条码属性页条宽比选项的默认设置为“3.00”，并允许应用程序选择最合适的条宽比(此比例值依赖于打印机的精度，不能连续调整)。

**码高** 用于指定条码的高度。使用条码属性页中的码高选项来指定从条码的顶边缘到底边缘条的长度。条码中最高的条长度不能比页的高度还高。条码中最短的条长度不能小于 1 毫米。

也可能通过鼠标拖动更改条码高度,但通过设置码高可以精确的设置条码的高度。

---

**关于矮条码的提示:** 应该尽量避免使用矮的条码。因为这可能造成扫描器扫描困难,降低条码的识读性能。

---

## 条码特殊选项

对于不同类型的条码，拥有不同的特殊选项。

**校验字符** 用于指定是否在条码中添加校验字符(适用于 Code39、ITF25、Codabar、Martrix 25 码等)。

条码属性页中的校验符选项允许指定是否将校验符添加到条码后面。校验符是一个添加到条码后面的可选字符，用来进行错误校验。

**为什么要使用校验符:** 尽管用条码识读器进行自动数据输入比使用键盘进行数据输入可靠性要高许多倍,但理论上仍不能排除解码错误的可能性。比如,在用户扫描动作中的意外抖动,或者条码的打印缺陷都可能导致扫描错误的发生。错误的出现会对数据库的完整性带来灾难性的后果,所以在这种情况下强烈建议使用校验符。尽管校验符并不能完全排除出现替换错误的可能性,但校验符确实可以降低出错率,从而保证数据的完整性。特别是当条码无法识读而用手工录入时,通过校验字符可检验输入的正确性。

**校验符是如何工作的:** 当条码识读器配置成检验校验符后,就可以扫描出包含校验符的条码,再重新执行校验符最开始生成时由应用程序所作的计算。条码识读器将计算得到的校验和与由应用程序打印出的校验符相比较。如果这两个数值相同,条码识读器就会传输条码信息。然而,如果两个数值不同,条码识读器就会知道发生了错误,条码字符不会被传输。用户必须重新扫描条码信息。当然,用户也可以在自己的应用程序中检验校验字符正确性。

**校验符的符号特别信息:** 尽管 Codabar 符号类型规范中并不包括

校验符，但一般仍可使用校验字符来提高条码的安全性，不同的是此种条码的校验字符正确性需要用用户程序自行检验。

校验符的类型有以下几种选择：

**模 43 校验** 这是打印 Code 39 条码最常用的校验符类型。这也是对 Code 39 条码唯一一种允许在数据字符串中包含非数字字符的校验符。当选择了其它校验符类型，应用程序则忽略条码数据字符串中所有的字母。

**模 10 校验** 这种校验符是使用与 ITF25、UPC、EAN 和 UCC/EAN-128 相同的校验公式计算出来的。但只有在条码数据字符串中只包含数字，才能选择这种设置。

**UCC/EAN-128** 用于指定输出符合 UCC/EAN-128 规范的条码。(只适用于 Code128。)

1991 年，Uniform Code Council (UCC) 和 International Article Numbering Association (EAN) 发布了用于全球范围内的零售、工业及贸易产品配送的新的条码标准，这些新标准定义了一个称作 UCC/EAN-128 的 Code 128 条码特别版本。LabelShop 完全支持 UCC/EAN-128 标准。

当条码属性页中的 UCC/EAN-128 选项被允许时(在检取框中包含检取标记)，应用程序自动添加某些字符并执行某些需要的功能，生成 UCC/EAN-128 条码。

当创建一个 UCC/EAN 128 条码时，可以在数据流中的任意位置通过输入符号(^)与(1)结合在一起组成的 FNC1 功能字符。即“^1”将插入一个功能符 1 (FNC1)

**字符集** 用于指定 Code 128 条码采用哪种字符集(自动、字符集 A、字符集 B、或字符集 C)。(只适用于 Code128。)

尽管 Code 128 码能对所有 128 个字符进行编码，但字符集本身只包括 102 个字符。Code 128 码实际包含三个不同的集合，每个集合都有 102 个字符，通过三个字符集的组合来达到表示 128 个字符的目的。任何 Code 128 码使用的默认字符集由起始符决定。然而，一旦一个 Code 128 码开始后，可以在条码中间切换不同的字符集。由于在单个条码中可以使用三个字符集的任意组合，当将数据编码进条码时，Code 128 码会给出最大的字符密度。

以下列出 Code 128 码的三个字符集：

**字符集 A** 全 ASCII 集，除了 26 个小写字母。

**字符集 B** 全 ASCII 集，除了 26 个控制字符。

**字符集 C** 双密度数字。这种字符集只包含数字，但任何一个条码符号实际代表两位数字。这样，字符集 C 中 102 个字符中有 100 个字符用于表示 100 个两位数字，从 00 到 99。

条码属性页中字符集选项的默认设置为“自动”。当设置为“自动”时，应用程序自动选择最有效的字符集，它可以出现在条码的开始，并可在字符集之间自动切换，以减少整个条码的尺寸。

**纠错级别** 用于设定二维条码的纠错级别。(只适用于 PDF417。)

**错误检测与错误纠正：**二维条码可对大量信息进行编码，但相对于更传统的（一维）条码，在扫描期间出现解码错误的可能性也会增加。因此，只使用简单的校验符不能满足需要，因为它只允许在扫描时检测出错误。由于这需要再次扫描，并且二维条码编码相当复杂，以至于在后来的扫描中可能会出现额外的错误，因此二维条码采用了高级的错误纠正功能。这是通过将足够的多余信息编码进打印的符号中，使扫描器不仅能检测出发生的错误，而且还能纠正这些错误。

**尺寸和可读性：**二维条码一般支持多个错误纠正等级。当纠正等级设置为低时，会编码进少量的多余信息，并且扫描器只能改正少量的错误。当纠正等级设置为最高时，可以改正严重的扫描问题。然而，随着错误纠正等级的增加，符号也会越来越大，因此，需要根据实际情况选择一个适合的纠错等级。

**如何决定错误纠正等级：**二维条码默认的错误纠正等级是能满足一般的场合。如果在一个封闭的系统里使用二维条码(就是说，能够既能控制打印质量，又能控制扫描质量)，并且对打印和识读能力很有信心，可以通过手动指定一个相对较低的错误纠正等级，减少条码符号的打印尺寸。另一方面，如果打印质量有问题，条码又用于比较恶劣的环境，或会有许多不同的人群使用各种各样的扫描设备，就会希望手动指定一个更高的纠正等级，即使知道打印出的符号会很大。

**截短型 PDF417** 用于指定采用 PDF417 条码的截短版本，以节省占用的空间。(只适用于 PDF417。)

当检取了截短型 PDF 417 选项时，条码符号右侧的终止符将被替换成一个竖条。尽管这样做在某种程度上节约了空间，但一般不推荐使用，除非打印质量和使用的扫描设备都能得到控制。

**层高** 用于指定 PDF417 条码每层的高度。(只适用于 PDF417。)

**列数** 用于指定 PDF417 条码数据区域中列的数目。(只适用于 PDF417。)

PDF417 条码符号中每层的高度默认设置为“X 尺寸”(最窄条的宽度)的三倍。对于编码数据, 如果使用错误纠正等级不小于 PDF417 条码符号规范标准推荐的最小值, 或使用“自动”错误纠正等级, 那么默认层高为三是比较合适的。然而, 如果手动指定错误纠正等级小于根据条码符号规范标准推荐的最小值, 推荐增加 PDF417 条码符号的层高。

**自动控制 PDF417 条码的形状:** PDF417 条码分成多层信息, 每层看上去好像是一个矮的条码。如果使这些层更接近, 就会注意到扩展到条码符号整个高度的最开始的五个条和最后面的五个条更象一个传统的条码。这些起始符和终止符有助于扫描器认出当前扫描出的条码符号类型是 PDF417。在这些起始符和终止符的中间, 编码的信息看上去有点象一个不规则的跳棋棋盘。当列数设置为“自动”, 在这个跳棋棋盘式的数据区域中的层数和列数会自动调整, 以便使整个条码的宽度与高度的比值接近 2:1。

**手动控制 PDF417 条码的形状:** 除了“自动”设置, 可以手动指定数据区域中的列数。每列是由四个条、四个空和定义编码数据的基础单位组成。每层最少由三列(12 个条和 12 个空)信息组成。这是因为在每层的左侧和右侧总包含一个指示符, 它可以告诉扫描器每层中其余数据的编码方法。在这两列(8 个条和 8 个空)之间最少有一列(4 个条和 4 个空)信息。在这部分数据区域中的列数可以通过列数选项设置, 范围是 1 到 30。

**符号宽度和高度:** 需要注意, 对于任何给定的错误纠正等级设置, 对同样数量的数据信息进行编码, 减少列的数量将增加层的数量。因此, 尽管可以选择减少列数而使 PDF417 符号比“自动”设置相对正方形尺寸更窄, 这样会使符号更高。当对大量数据进行编码时, 减少列数会导致 PDF417 码太高, 而不能放入标签中。

**起始符** 用于指定 Codabar 条码从四个字母, a - d 中选择哪一个作为起始符。(只适用于 Codabar。)

**终止符** 用于指定 Codabar 条码从四个字母, a - d 中选择哪一个作为终止符。(只适用于 Codabar。)

**附加条码** 用于指定是否在 UPC 或 EAN 条码后面使用附加条码。(只适用于 UPC-A、UPC-E、EAN-13、和 EAN-8。)

## 供人识读字符

**位置** 条码属性页中的放置位置选项允许在条码的下面或上面放置供人识读字符。供人识读字符放置位置的默认设置是“条码下方”。

---

**注意：**EAN/UPC 条码的供人识读字符不能进行人为控制，只能选择默认或无。二维条码没有供人识读的字符。

---

**对齐方式** 选项用于指定供人识读字符与相应条码之间的排列方式：左齐、右齐、居中和撑满。

**左齐** 供人识读字符的左边缘与条码最左边条的边缘对齐。

**右齐** 供人识读字符与条码最右边条的边缘对齐。

**居中** 供人识读字符在条码左右两个边缘之间居中放置。

**撑满** 供人识读字符一个与另一个并排或重叠排开，第一个字符的左边缘与条码最左边条的边缘对齐，最后一个字符的右边缘与条码最右边条的边缘对齐。如果供人识读字符的长度小于条码的长度，字符之间会有间距；如果供人识读字符的长度大于条码的长度，字符之间会重叠放置。

**垂直偏移** 选项用于指定供人识读字符与条码之间的垂直距离。

**字符模板** 格式化条码的供人识读的文字。

我们会常常遇到，条码应用程序需要在条码供人识读字符中打印出一些额外的字符，但不是扫描出条码中的一部分。比如，在一些大型零售商店需要的 UCC-128 货运包装箱标签上，需要一些额外的空格和括号等字符。这些字符并不是条码的一部分，也就不会被条码识读器读入到计算机。在条码属性页中的字符模板选项用于指定数据字符串中这些额外的字符和它们的位置。

以下是一个条码供人识读字符的简单例子。字符模板为：“(??) ???-???”，条码包含数字字符串“1234567890”，显示结果为：“(123) 456-7890”。字符模板选项中的每个问号(?)表示实际条码数据字符串中的一个字符。“()”、“-”等符号代表将出现在供人识读字符中的额外数据，这些字符并不是条码符号的一部分。

## 文本属性页

当在标签中创建文字对象时，总是默认为单行文字对象。然而，一旦在标签中放置了文字对象，可以指定多种附加的格式特性，包括将文字变



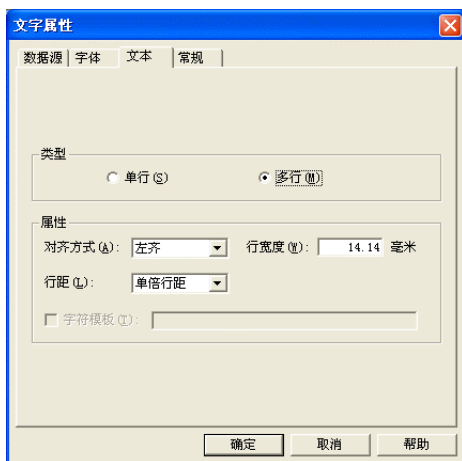
成多行。(注意这里所说的格式特性是指文字对象整体的形状，而不是字体特征本身，如粗体、斜体、字体名称和字体尺寸。)

文本属性页中的选项会根据指定的文本类型而改变：

**单行** 指定文字对象显示为单行。

**多行** 指定文字对象如果超出行宽度，则自动换行。

当指定了文本属性页中的多行选项，一些选项就允许选择，如对齐方式、行宽度和行距。



与多行选项相关的选项列出如下：

**对齐方式** 用于指定多行文字的对齐方式是哪一种：左齐、右齐和居中。

**行宽度** 用于指定文本行的宽度。

更改选定对象的行宽度的最容易方式是通过鼠标拖拽对象左边或右边的句柄。也可以使用文本属性页中的行宽度选项。

---

**重要提示：**选定的多行文字对象只有两个激活的句柄可用，即对象左右两边中间的句柄。这些句柄只能用于修改对象块本身的尺寸。如果要改变对象中文字的属性，必须使用字体属性页。

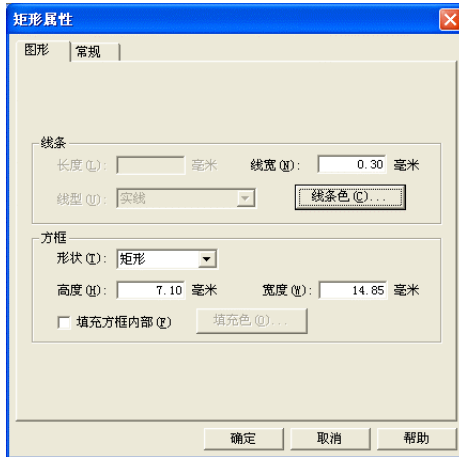
---

**行距** 用于指定多行文字每行之间的间距，如果选择了自定义行距，会有一个输入框用于输入行距数值。

**字符模板** 指定文字对象是否使用字符模板，并可用来定义字符模板。其含义和用法与条码属性页中的字符模板完全相同。

## 图形属性页

图形属性页用于修改标签中矩形和线条对象的高度、宽度和线宽等。图形属性页还包含线条色按钮和填充色按钮，用于显示颜色对话框并更改矩形的边框和内部颜色。



图形属性页中的选项列出如下：

**高度** 用于指定矩形的垂直尺寸。

**宽度** 用于指定矩形的水平尺寸。

**线宽** 用于指定矩形边框线条的宽度。

**长度** 用于指定线条两端的长度。

**填充方框内部** 用于指定是否要将矩形内部用实颜色填充。

**线条色** 用于指定矩形边框的颜色。

**填充色** 用于指定矩形内部的颜色。

**线宽** 用于指定标签中线条或矩形边框的宽度。默认值为 0.30 毫米。

**线条长度** 用于指定线条两端的长度。指定线条长度最容易的方法是使

用鼠标。然而，也可以通过在图形属性页的长度选项中填入数值的方法指定线条的长度。这两种方法都在以下说明。

#### 使用鼠标的方法：

1. 单击要更改长度的线条对象。
2. 将鼠标指针移动到线条两端的句柄上，单击并一直按住鼠标的左键。
3. 按住鼠标左键，并拖拽选定的句柄直到线条达到要求的长度。
4. 释放鼠标左键。

#### 使用属性页的方法：

1. 双击要更改长度的线条对象，显示出直线属性对话框。
2. 在相应的属性页中选择长度选项。
3. 在编辑框中输入需要的线条长度数值。
4. 单击确定按钮。

**方框的高度和宽度** 用于指定矩形高度和宽度。指定矩形高度和宽度同样有两种方法：

#### 使用鼠标：

1. 单击要更改尺寸的矩形对象。
2. 将鼠标指针移动到矩形的句柄上，单击并一直按住鼠标的左键。
3. 按住鼠标左键，并拖拽选定的句柄直到矩形达到要求的宽度。
4. 释放鼠标左键。

#### 使用属性页：

1. 双击要更改宽度的矩形对象，显示出矩形属性对话框。
2. 在相应的属性页中选择宽度选项。
3. 在编辑框中输入需要的矩形宽度数值。
4. 单击确定按钮。

---

**注意：**可以使用边角的句柄同时调整矩形的高度和宽度。当指定标签中线条长度和矩形宽度时，如果指定的矩形宽度使矩形边框超出标签查看

---

区域，可能只有在标签边界以内的部分才能打印出来。

**填充方框内部** 用于将当前选定的矩形对象内部填充为实色。默认的填充色为黑色。然而，可以使用颜色对话框将矩形框内部填充为任何可用的颜色。

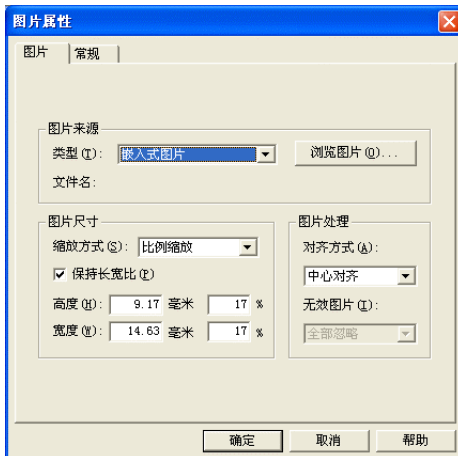
**注意：**并不是所有的打印机都支持填充功能。

## 图片属性页

图片属性页包含了一些选项，用于选择引入哪个图片和如何引入图片，调整图片的尺寸以适合标签，还决定如果在打印时未找到图片该如何进行处理。

当第一次使用引入图片按钮将一个图片插入到标签中时，本应用程序最初插入一个默认图片。这个默认的图片实际只是起到占位符的作用。通常都是要通过图片属性页指定一个有效的图片文件。

要调整图片的位置，使用鼠标或常规属性页中的选项。



### 图片来源选项：

**类型** 用于指定本应用程序如何引入图片到标签格式中。这个选项允许

嵌入、链接或从外部数据文件读取。如果选择了“数据源图片”，会出现数据源属性页，在数据源属性页中指定图片名称的获取方法。

**浏览图片** 用于指定要插入标签格式中的图片文件。(只适用于嵌入式图片和链接式图片。)

**文件名** 用于显示要引入标签格式中图片的路径和文件名。(只适用于链接式图片)。

数据源图片一般用于制作证卡等领域。可以将图片预先扫描完成，然后建立一个数据文件，其中的各个字段分别是图片名称和一些与此图片相关的信息(如姓名等)。在标签格式中引入该数据文件，将图片设置为数据源图片格式，并在数据源属性中选择数据库数据类型。一般来讲，将标签格式文件和图片文件放在同一目录中是一个好的方法，LabelShop会自动到标签格式文档所在的目录中寻找未指定全路径的数据源图片。而对于链接式图片，则只按图片所在的原始路径查找。因此，如果改变了图片文件位置，请重新选择图片。

对于一些图标，采用嵌入方式将图标数据保存在标签格式文档中是一种不错的选择，这样不会因为丢失了图标文件而造成图标无法正常输出。

### 图片尺寸选项：

**缩放方式** 用于决定图片是否在标签查看区域可变尺寸，如果可变尺寸，以哪种方式改变。

**原始尺寸** 选项用来保持原始图片的尺寸，使之与引入时原始图片文件中图象的大小相同。当选择了该选项，图片的尺寸不能再修改。要想改变图片的尺寸，将缩放方式变为比例缩放或适合边框。

**比例缩放** 选项用于按照图片原始尺寸的百分比增加或减少图片的大小。使用鼠标调整图片的尺寸，或在高度和宽度后面的文本框中输入百分比。

**适合边框** 选项用于增大或缩小图片，使之强制显示或打印在指定区域。这个特性对于使用数据源图片特别有用。由于图片是通过数据库连接到应用程序的，它的尺寸可能会发生变化，适合边框选项保证从数据库引入的图片只按事先调整的尺寸显示或打印在标签中，而不是图片原来的尺寸。

---

**注意：**当图片选择适合边框属性，且图片数据源为数据库图片时，连续改变数据库记录查看标签，可能会使图片尺寸越来越小。

---

**保持长宽比** 选项用于设置是否允许使用按比例更改图片尺寸的能力。

可以使用图片属性页中的高度和宽度按比例更改图片尺寸。由于图片的尺寸是按比例显示或打印的，任何对图片高度的改变，都会引起宽度的变化；同样任何对图片宽度的改变，也会引起高度的变化。

允许使用鼠标按照需要改变并拉伸图片。也可以在图片属性页中使用高度和宽度选项改变或拉伸图片的尺寸。

**高度和宽度** 选项用于使用输入具体的数值和与原来图片的百分比来调整图片的尺寸。有两个数据输入框用来输入高度和宽度的具体数值，还有两个用来输入百分比。根据选择的缩放方式，应用程序将允许或禁止其中的一些或所有高度和宽度数据输入框。

### 图片处理选项：

**无效图片** 用于决定在指定的文件夹下找不到链接图片或数据源图片时，应用程序如何处理。

当使用链接式图片或从外部数据文件引入图片，应用程序查找保存在硬盘或软盘上的图片文件。**LabelShop** 首先按照图片文件的路径查找图片，如果没有找到，则会继续在标签格式文档所在的路径下查找图片。如果应用程序在打印时无法找到需要的文件，图片属性页中的无效图片处理选项用来选择应用程序如何处理打印作业。

### 无效图片处理选项：

**终止打印** 通知应用程序在无法找到图片文件时不打印标签。

**询问** 通知应用程序在无法找到图片时询问是否继续打印标签。

**打印标记** 通知应用程序在无法找到图片文件时不终止打印，而在图片的位置打印出固定的标记。

**全部忽略** 通知应用程序即使无法找到图片也继续打印。

**对齐方式** 用于决定不同尺寸的图片的对齐方式。当使用链接式图片或从外部数据文件引入图片时，由于每个图片的尺寸可能会发生变化。图片对齐方式选项用于指定如何安排不同尺寸图片的位置。

**中心对齐** 图片中心的水平和垂直方向都在指定区域各自的中心。

**左上角对齐** 图片的左上角在指定区域的左上角。

**上中对齐** 图片的水平中心在指定区域的水平中心，而图片的上边缘在指定区域的上边缘。

**右上角对齐** 图片的右上角在指定区域的右上角。

**右中**对齐 图片的垂直中心在指定区域的垂直中心,而图片的右边缘在指定区域的右边缘。

**右下角**对齐 图片的右下角在指定区域的右下角。

**下中**对齐 图片的水平中心在指定区域的水平中心,而图片的下边缘在指定区域的下边缘。

**左下角**对齐 图片的左下角在指定区域的左下角。

**左中**对齐 图片的垂直中心在指定区域的垂直中心,而图片的左边缘在指定区域的左边缘。

---

**特别说明:** 当使用数据源图片时,如果没有明确指定图片的路径,则程序会到当前路径和标签格式文件所在的路径下查找相应的图片。

---

## 对象属性的附加说明

当多个对象被同时选中时,通过属性页只能修改参考对象(也就是第一个对象)的属性。而利用工具栏和菜单,则可以同时更改所有选中对象的相应的属性(如字体、颜色等)。





## 第六章

# 对象的数据源

## 概述

通常，条码或者文字的内容称为数据，数据有时需要由多个部分组成，每个部分表示不同的内容，如一个电话号码可能包含区号、电话号码和分机号等。在 Labelshop 中可将数据分为若干段，每段可以表示不同的内容，并拥有独立的属性，称为子串。用户可以对各个子串进行单独的编辑和设置。子串又称为变量，变量有常量、数据库、序列号等类型，分别有不同的数据获取方式。如果在同一个标签格式中，多个对象的数据需要引用同一个变量，可以赋与这个变量一个名称，对象中只要通过名称引用这个变量就可以，这样的变量称为命名变量或共享变量。当共享变量发生变化时，与其联接的全部对象都会自动更新。

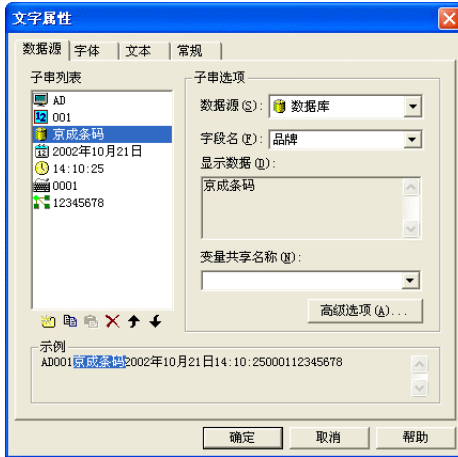
对于图片对象，如果在类型中选取了“数据源图片”，也会出现数据源属性供设置。图片对象的数据源用于提供图片文件的文件名。如果文件名中包含了完整的路径信息，则软件到指定的路径下寻找图片文件，如果未指定路径信息或在相应的路径下未找到图片，则程序会在文档所在的路径下继续寻找图片文件。

变量的类型有常量、序列号、数据库、日期、时间、键盘输入和脚本可供选择，每种类型的变量都有不同的数据获取方式和不同的选项设置。

软件可对变量进行一些格式化操作，如截短等，这些操作都是针对一个变量进行的。一些对象（主要是条码对象）对于数据的长度、内容有一定的要求（如 ITF25 码只能接受偶数位的数字），并对数据进行处理以

满足这些要求。对于一个对象的数据而言,首先是对每个变量进行取值,并进行相应的格式化处理,再将所有的子串连接在一起,最后针对整个数据进行格式化。在这里,会对每个变量的格式化是由用户决定的,而会对整个数据的格式化处理是由每个对象的特性决定的。

数据源属性页用于设置对象的数据源属性。



**子串列表** 显示组成对象数据的变量组合。可在此设置当前变量。

**子串选项** 设置当前选中的变量的属性。

**数据源** 设置当前变量的类型。

**显示数据** 是当前变量在标签中的显示内容,也用于输入初始数据,在实际输出时此内容可能会被具体的数据替换。

**变量共享名称** 显示此变量用于共享时的名称,此名称不区分大小写。

**高级选项** 设置变量的一些附加设置。

**示例** 显示此对象的数据示例。

如何对多个子串进行操作,请参考后面的多子串操作和子串列表。

每个变量的属性可被单独设置,命名共享变量的属性更改后,其它与之相联的对象也会同时被更新。不同类型的变量拥有不同的属性设置,子串选项区域会根据变量类型的不同显示相应的设置选项。有此类型的设置信息需要在高级选项中进行,请参考下面每种变量类型的具体信息。

## 常量

常量数据类型对于每个标签都是固定的内容。对于新创建的对象，应用程序默认的数据源类型设置总是常量。

只要直接在显示数据选项中输入需要的数据，就可以完成常量的输入。

常量数据在所有的标签中都是固定的值，适用于固定的数据对象。

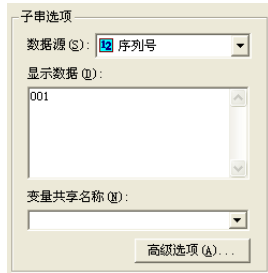


## 序列号

“序列号”也称为“计数器”，其内容是根据标签的顺序号有规律进行变化的。

对于序列号数据类型，显示数据是序列号的起始数据。只要直接在显示数据输入框中输入需要的数据，就可以完成序列号起始数值的输入。

序列号有多种类型，每种类型都有一个字符集序列。初始数据中从右侧起符合字符集序列的字符都可以被序列化。



序列号的属性在高级属性中的序列号属性页中设置。

### 序列号属性页

序列号属性页用于指定如何对选择的数据子串进行序列化。

**类型选项** 用于指定序列号子串数据使用的字符集。“10 进制（数字）”是默认设置，它适合大多数应用环境。也可以指定其它设置。

可供选择的类型如下：

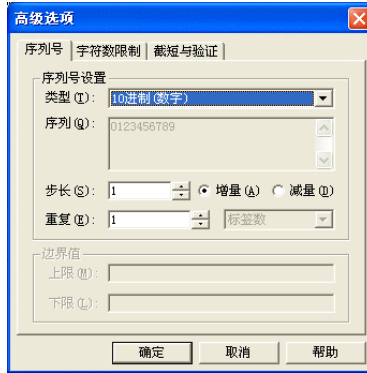
**10 进制（数字）** 只由 0~9 十个数字组成的字符集。

**26 进制（字母）** 由 A~Z 二十六个字母组成的字符集。尽管在这个字符集中有 26 个字符，但不是象数字系统那样从 0 开始。

**36 进制（数字和字母）** 由 0~9 十个数字和 A~Z 二十六个字母。

**16 进制 (0-9, A-F)** 由 0~9 十个数字和 A~F 六个字母。

**序列** 根据选择的类型显示字符集的所有字符排列。序列字符的变化依赖于这个字符序列的顺序。



**增量/减量** 指定是增加还是减少子串数据。同时指定数据更新的数值。

增量选项用于指定子串数据使用增量方式;而减量选项用于指定子串数据使用减量方式。步长选项用于填入子串数据增加或减少的数值。

这些选项的默认设置为“增量”，步长为“1”。对于大多数应用，这些设置都工作得很好。举一个例子，假设需要子串数据每个标签减少2。应该更改设置为“减量”，步长为“2”。

**边界** 指定序列号的最大和最小数值。

边界值选项用于指定最大或最小数值。数据的增加或减少后的数值不能超出这个范围。上限选项用于输入数据增加后允许的最大值，下限选项用于输入数据减少后允许的最小值。

例如，子串数据增加到 1000 的时候，重新从 1 开始。将上限选项设置为 1000。

## 数据库

在一个打印作业中，如果需要打印许多标签，并且每个标签的数据都是变化的，“数据库”是首选的数据源类型。如果用户数据保存在数据表格中，也可用此数据类型进行输出。

由于这些数据对于每个标签通常是不同的，文件中的数据只有当前记录

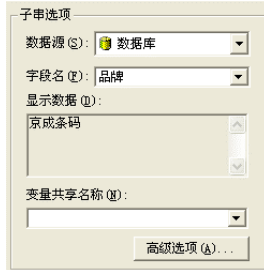
显示在标签查看区域。可以通过选择数据库菜单中的移动当前记录命令或单击数据库工具栏中的移动记录按钮显示其它记录的数据。

**字段名** 用于指定数据库中的哪个字段打印时替换当前指定的字符串示例数据。

**显示数据** 用于指定显示在标签打印区域中的数据。在打印时，这个示例数据会由数据源文件中的数据替换标签中的内容。

“数据库”方式可将用户数据库中的内容输出到标签上，同一标签上可设置多个记录字段，但同一标签上的各字段的数据取自同一记录。

后面有专门的章节说明如何在标签格式中使用数据库联接。

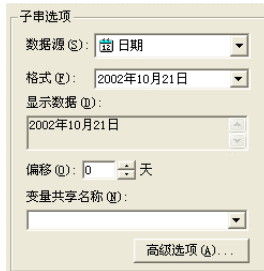


## 日期

“日期”数据类型将系统当前的日期作变量内容，可以选择多种输出格式，并可设置日期偏移。

**日期格式** 指定日期输出格式。

**偏移** 用于指定要显示的日期与当前日期的偏移。正值指定未来的（推迟）日期，而负值指定过去的（提前）日期。

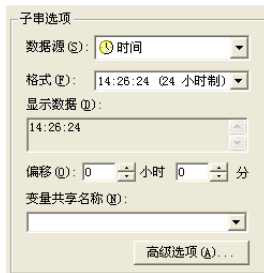


## 时间

“时间”数据类型将系统当前的时间作变量内容，可以选择多种输出格式，并可设置时间偏移。

**时间格式** 用于指定时间变量的输出格式。

**偏移** 用于指定要显示的时间与当前时间的偏移。正值指定未来的（推迟）时间，而负值指定过去的（提前）时间。



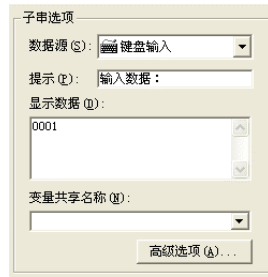
日期和时间的数据取自于当前系统的日期和时间，偏移也是以当前系统的日期和时间为基准的。显示的日期时间只是一个参考值，实际输出的日期时间是依赖于输出时的日期和时间的，因此同一页标签上相同的时间也会由于处理输出的时刻不同而存在差异。

**日期时间偏移示例：**假设当前的日期是 2002 年 3 月 12 日，时间是 12 时 22 分。需要打印的标签上，需要失效期是 2002 年 3 月 15 日，生产时间是 12 时 22 分。需要在时间的偏移选项中输入“3”，时间将显示为 2002 年 3 月 15 日。也就是说，如果需要输出的时间不是当前日期，而是一个与当前日期有一个固定偏移的日期（如药品的失效期），则应当使用偏移。

## 键盘输入

当标签的信息只有在打印时才能确定时，可以使用“键盘输入”数据类型，以通知应用程序在打印作业的开始时请求输入数据。例如，可能需要打印标准的货运标签，其中包括包裹的重量信息，而且包裹的重量只有在打印的时候才能知道。

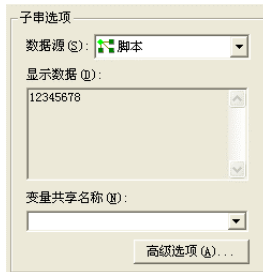
提示信息用于在要求用户输入数据时显示提示信息。



当需要打印标签时，会出现一个要求用户输入数据的对话框，当用户输入了新的数据后，显示数据将被新数据所找替，在所有输出的标签中，键盘输入类型变量的数据都是相同的。

## 脚本

LabelShop 支持 VBScript 脚本语言用于数据的获取。用户可以编写自己的脚本用户格式化数据、计算校验字符、根据标签的位置计算变量数据、进行自己的数据验证等，甚至可以用脚本打开数据库并检索出数据，同时控制 LabelShop 进行输出。LabelShop 负责提供相应的环境并传递必要的数。



由于脚本是一种复杂的高级应用，因此要求用户对脚本相当熟悉。如果不是这样，建议您不要使用脚本功能。

---

**注意：**下面的内容并不是 VB 脚本的语言参考，而是关于 LabelShop 如何支持脚本的一些技术细节，以便于用户能正确地在 LabelShop 中使用脚本语言。

---

#### 在 LabelShop 中使用脚本应当注意：

- 1、 LabelShop 默认是不运行脚本的，如果想运行脚本，请在系统设置中选中“允许脚本运行”选项。
- 2、 LabelShop 支持 VBScript 脚本语言。
- 3、 LabelShop 通过调用几个过程来完成数据获取工作，所得到的数据由用户脚本决定。
- 4、 LabelShop 在更新显示数据、打印预览和打印输出时会运行脚本，但不同情况下运行脚本的方式也不相同，LabelShop 会传递相应的信息供用户区分不同的环境。
- 5、 LabelShop 还向用户脚本传递如标签所在的页号、行号、文档名称以及命名变量的数据，供用户脚本参考使用。
- 6、 脚本有三种类型：系统定义脚本、文档全局脚本和对象私有脚本。系统定义脚本是 LabelShop 提供的一组脚本函数，可以完成一些常用的校验位计算；文档全局脚本中包括两个由 LabelShop 调用的函数，函数实现的功能由用户定义，LabelShop 在开始和结束执行脚本时会调用这两个过程；私有脚本适用于当有变量，LabelShop 调用其中的一个函数得到变量的数据。

#### LabelShop 中使用脚本的约定：

- 1、 在文档全局脚本中，系统预定义了四个函数：**Function OnBeginPrint(State)**和 **Function OnEndPrint**，这两个函数是由 LabelShop 调用的。State 传递的是运行脚本的环境，0 表示是在更新显示数据时运行的脚本，1 表示是在打印预览时运行的脚本，2 表示是在打印输出时运行的脚本；**Function OnBeginLabel** 和 **Function OnEndLabel** 函数分别在输出每个标签前和每个标签输出完成后被调用。函数的返回值没有意义。
- 2、 在对象的私有脚本中，**OnGetData** 是由 LabelShop 调用来获取数据的，函数的返回值就是变量的数据。
- 3、 系统提供了几个全局变量，用于传递一些环境信息，用户可以在脚本中使用这些变量。
- 4、 在 **OnBeginPrint(State)**函数中，用户可修改 **V\_TOTALLABELS** 全

局变量，直接控制输出标签的数量。

- 5、文档中的命名变量可在脚本中引用，但变量的名称必须符合规范。在 LabelShop 中，要在脚本中引用的变量不能是一个脚本类型的变量，且名称必须由字母、数字和下划线组成，且第一个字符必须是字母。

LabelShop 提供了如下全局变量：

V_PAGE	标签所在的页号（基于 1）
V_COPY	当前标签的拷贝数量
V_ROW	标签所在的行号（基于 1）
V_COL	标签所在的列号（基于 1）
V_LABELNO	标签的序号（基于 1）
V_TOTALLABELS	总计输出的标签数（基于 1）
V_STARTNO	标签的起始输出位置（基于 1）
V_TITLE	文档名称
V_PRINTER	用于输出的打印机名称

用户可在自己的脚本中使用和修改这些全局变量，但只有部分修改是有效的。除了在预览和打印时修改 V\_TOTALLABELS 变量以重新定义输出标签数量外，不建议修改其它全局变量。另外，在特定的情况下，上面的一部分变量的值可能没有意义。

在不同的环境下，全局变量的不同点如下：

- 1、当更新显示数据时：当重新更改了脚本或脚本中引用的变量值发生变化时，LabelShop 要运行脚本更新变量的数据。

LabelShop 首先调用 OnBeginPrint(0)，再调用 OnBeginLabel，然后调用相应变量的 OnGetData（可能会调用多个变量的函数），最后调用 OnEndLabel 和 OnEndPrint。在此情况下，V\_PAGE、V\_COPY、V\_ROW、V\_COL、V\_LABELNO、V\_TOTALLABELS、V\_STARTNO 七个全局变量的值均被设定为 1，并且不建议您在这种情况下修改 V\_TOTALLABELS 变量的值，V\_TITLE 和 V\_PRINTER 的内容根据文档的内容而定。

- 2、当执行打印预览和输出时：LabelShop 要运行脚本获取变量数据。



LabelShop 首先调用 OnBeginPrint(1) (预览时) 或 OnBeginPrint(2) (打印时), 您可在该函数中修改 V\_TOTALLABELS 全局变量的值, LabelShop 将根据您设定的值进行输出。如果您不对其进行设定, 则仍使用打印对话框中设定的数值。如果为 V\_TOTALLABELS 设置一个小于或等于 0 的值, 程序将不进行预览或打印输出。对于每个标签, 在标签输出前, LabelShop 会设定 V\_PAGE、V\_COPY、V\_ROW、V\_COL、V\_LABELNO 的值, 并调用 OnBeginLabel, 然后调用标签中每个脚本变量的 OnGetData 函数, 再调用 OnEndLabel 完成一个标签的输出过程。在所有标签输出完成后调用 OnEndPrint 函数。在标准的图形输出方式下, 所有的全局变量的值都是有意义的, 而在命令输出方式下, 在此情况下, V\_PAGE、V\_ROW、V\_STARTNO 三个全局变量的值均被设定为 1, 没有实际意义。V\_TITLE 和 V\_PRINTER 的内容根据文档的内容而定。

- 3、符合 LabelShop 脚本变量命名规则的命名变量的值也被做为全局变量传递!
- 4、V\_PAGE、V\_COPY、V\_ROW、V\_COL、V\_LABELNO、V\_TOTALLABELS、V\_STARTNO 全局变量的值是以数字型传递的, 其它全局变量和相应的命名变量的值是以字符串型传递的。

#### LabelShop 运行脚本的方式:

假设有如下标签格式: 标签纸为 JC-8R, 格式文档名称为 “TestScript”, 打印机为 “Lexmark T620” 标签上有两个文字对象, 对象 1 的变量为序列号类型, 步长为 1, 重复数为 1, 共享名称为 “SN”, 初始值为 “1001”; 对象 2 的变量有脚本型, 共享名称为 “SN”, 私有脚本为:

```
Function OnGetData ()
    OnGetData = s & SN + V_PAGE * 10 + V_ROW
End Function
```

文档的全局脚本为:

```
Dim s
Function OnBeginPrint(State)
'State = 0 : Variable UpdateData
'State = 1 : Label Preview
'State = 2 : Label Print
    if (State = 0) Then
```

```
        s = "SC = "  
    else  
        s = ""  
    end if  
End Function  
  
Function OnBeginLabel()  
End Function  
  
Function OnEndLabel()  
End Function  
  
Function OnEndPrint()  
End Function
```

在更新变量的情况下, LabelShop 首先加工出如下脚本:

```
'全局变量  
V_PAGE = 1  
V_COPY = 1  
V_ROW = 1  
V_COL = 1  
V_LABELNO = 1  
V_TOTALLABELS = 1  
V_STARTNO = 1  
V_TITLE = "TestScript"  
V_PRINTER = "Lexmark T620"  
  
'命名变量  
SN = "1001"
```

'此处的函数和过程可供所有的变量调用

..... (系统定义脚本)

'全局脚本

Dim s

Function OnBeginPrint(State)

'State = 0 : Variable UpdateData

'State = 1 : Label Preview

'State = 2 : Label Print

if (State = 0) Then

s = "显示"

else

s = ""

end if

End Function

Function OnBeginLabel()

End Function

Function OnEndLabel()

End Function

Function OnEndPrint()

End Function

然后调用 OnBeginPrint(0), 再调用 OnBeginLabel;

再加入脚本:

Function OnGetData()

OnGetData = s & SN + V\_PAGE \* 10 + V\_ROW

```
End Function
```

并且调用 `OnGetData`，并将函数返回值赋给 `SC` 的变量，如果标签格式中有多个脚本变量需要更新，`LabelShop` 会再次加入下一个变量的私有脚本（私有脚本中的同名变量和函数会自动覆盖上次传入的内容），并再次调用 `OnGetData`。

全部变量更新完成后，调用 `OnEndLabel` 和 `OnEndPrint`，对象 2 的数据被更新为“`SC = 1012`”

在预览和打印输出的情况下，假设打印数量设定为 10，起始标签为默认值，`LabelShop` 首先加工出如下脚本：

```
'全局变量
V_PAGE = 1
V_COPY = 1
V_ROW = 1
V_COL = 1
V_LABELNO = 1
V_TOTALLABELS = 10
V_STARTNO = 1
V_TITLE = "TestScript"
V_PRINTER = "Lexmark T620"
'命名变量
SN = "1001"
'此处的函数和过程可供所有的变量调用
.....（系统定义脚本）
'全局脚本
Dim s
Function OnBeginPrint(State)
'State = 0 : Variable UpdateData
'State = 1 : Label Preview
```

```
'State = 2 : Label Print
  if (State = 0) Then
    s = "显示"
  else
    s = ""
  end if
End Function
```

```
Function OnBeginLabel()
End Function
```

```
Function OnEndLabel()
End Function
```

```
Function OnEndPrint()
End Function
```

然后调用 OnBeginPrint(1) (预览) 或 OnBeginPrint(2) (打印输出);

然后对于每个标签, 重新设置 V\_PAGE、V\_COPY、V\_ROW、V\_COL、V\_LABELNO 全局变量, 并重新设置命名各变量, 例如对于第 4 个标签:

再加入如下脚本:

```
V_PAGE = 1
V_COPY = 1
V_ROW = 2
V_COL = 2
V_LABELNO = 4
'命名变量
SN = "1004"
```

调用 OnBeginLabel,

对于 SC 变量, 再加入:

```
Function OnGetData()
    OnGetData = s & SN + V_PAGE * 10 + V_ROW
End Function
```

对于 SC 变量调用 OnGetData, 返回值为 “1016”。

调用 OnEndLabel。

对于全部标签, 重复这个过程。

全部标签输出完成后, 调用 OnEndPrint 函数。

**需要特别注意的是:** 在打印输出时, 标签的顺序一般是从 1 开始递增的, 而在预览时则不同, 因为可能从后一页转到前一页进行预览, 因此标签号可能无序变化, 同一标签的脚本也可能被多次执行。

脚本的内容需要在高级选项中的脚本属性页中输入

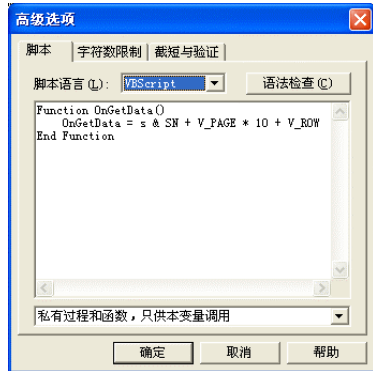
**脚本语言** 选项用于选择脚本的类型, 目前只支持 VBScript。

**语法检查** 用于检验脚本有无语法错误。

最下面的列表框用于选择当前编辑的脚本的是私有的还是全局的。

输入框用于输入脚本的内容, 系统定义的脚本是无法更改的。

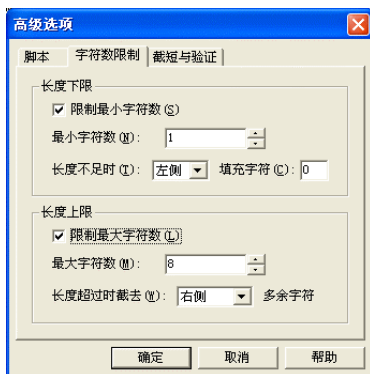
如果脚本中有错误, 语法错误可用语法检查检验, 而其它错误只有在执行时才能发现。LabelShop 还不能对脚本中的错误返回出错信息, 只是简单地脚本变量设置一个空的字符串。因此需要用户自己确保脚本的正确性。建议首先在 VB 中将脚本调试正确后, 再复制到 LabelShop 中。如果由于脚本中的错误 (如死循环等) 造成 LabelShop 停止响应, 您只能手工强制中止 LabelShop 运行后, 再重新运行 LabelShop, 并在系统设置中设置不允许运行脚本, 然后打开含有错误脚本的文档, 并更正错误, 再更改系统设置, 恢复脚本执行。



## 其它高级选项

### 字符数限制

使用字符数限制属性，可以限制对象数据的长度。



#### 长度下限

有时从数据文件某字段读取的数据中包含的字符数与打印条码或文字对象需要的字符数少。当出现这种情况，在字符数限制属性页中的长度下限选项就用于如何处理这些数据。

长度下限选项描述如下。

**限制最小字符数** 用于指定是否允许使用最小字符数选项。

**最小字符数** 用于指定从数据文件读取的最小字符数。

**填充** 用于指定在字符长度不足时，是在数据的左侧还是右侧填加少于最小长度数量的字符。

**填充字符** 用于指定填充时使用的字符。

#### 长度上限

有时从数据文件某字段读取的数据中包含的字符数与打印条码或文字对象需要的字符数多。当出现这种情况，在字符数限制属性页中的长度上限选项就用于如何处理这些数据。

长度上限选项描述如下。

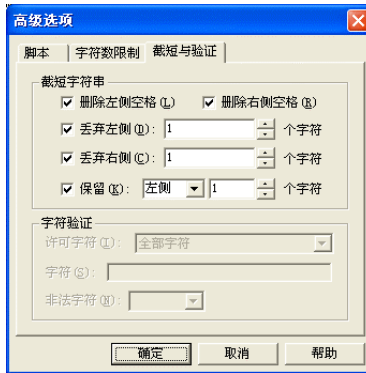
**限制最大字符数** 用于指定是否允许使用最大字符数选项。

**最大字符数** 用于指定从数据文件读取的最大字符数。

**截去** 用于指定在字符长度超过时,是在数据的左侧还是右侧截去多余字符。

## 截短与验证

截短选项用于指定哪些字符保留或删除或用一定的方式进行替换。



当从数据源文件读取数据时,有时可能不需要将数据文件字段中的整个内容读入到标签对象。例如,可能数据文件字段在需要的数据前面或后面包含多余的空格。(这可能是由于从一种数据库格式转换到另一种数据库格式,或从数据库转换到文本文件时引起的。)由于应用程序无法判断这些空格是否有必要,在打印时会同正常数据一样看待。这就会增加条码的长度并引起其它问题。

**删除左侧空格** 用于删除开始不需要的空格。

**删除右侧空格** 用于删除结尾不需要的空格。

**丢弃左侧字符** 用于删除数据文件字段从左边开始的指定字符数。

**丢弃右侧字符** 用于删除数据文件字段从右边开始的指定字符数。

**保留左侧字符** 用于保留数据文件字段从左边开始的指定字符数。

**保留右侧字符** 用于保留数据文件字段从右边开始的指定字符数。

验证选项用于强制数据遵从某些规则。例如,可能会脱去不希望的数据。



字符验证选项列出如下。

**许可字符** 用于指定可以接受的字符类型。

**字符** 用于在许可字符选项设置为“自定义”时，指定可以接受的字符。

当同时选取了多个处理方式后，系统按以下顺序对数据进行处理：

对于一个给定的数据源，如果截短与验证与字符数限制的中的下列功能全部或者部分选中，其处理的顺序如下：

- 1、删除左侧空格
- 2、删除右侧空格
- 3、丢弃左侧若干个字符
- 4、丢弃右侧若干个字符
- 5、保留左侧/右侧若干个字符
- 6、长度下限
- 7、长度上限

汉字的特殊性在于一个汉字占用 2 个 ASCII 字符长度，设定丢弃或者保留字符可能会遇到处理半个汉字的情况，如果丢弃或者保留边界的位置恰好遇到半个汉字的时候，将自动丢弃剩下的半个汉字，实际舍弃的字符数量比设定的数量多 1 个。

## 子串列表与多子串操作

多数情况下，标签中的对象使用单一的数据源。然而有时会希望将两个或多个数据源的信息放置到单一的条码或文字对象中。可以通过将多个数据子串结合在一起，并使它们作为一个字符串输出（这种情况叫字符串的“串联”）。

数据源属性页中的子串列表选项用于指定子串，以便对子串进行类型定义和更改数据子串顺序。

以下主题包含关于了解和使用子串列表的重要信息。

### 使用子串列表可以达到两个目的

**显示数据源结构** 子串列表提供清晰、直观的描述，从中可以看到标签对象中有多少个子串，每个子串使用的数据源类型。每行列表左边的小图标代表子串使用的数据类型。而每行列表的右边是子串当前指定的样

本数据。

**编辑数据源结构** 子串列表也可以对每个子串的内容进行编辑,并更改子串的顺序。每次用鼠标单击列表中的一个子串,或用上下箭头更改选定子串的位置,新近选定的子串的重要信息也显示在列表中。这样可以快速查看并编辑每个子串的现有配置。

## 建立子串列表

当第一次显示给定标签对象的数据源对话框,在子串列表中只显示一个子串。这个子串数据的示例数据显示在列表中顶部第一行,并反白显示。如果选择的对象使用常量数据源类型,无论当前在显示数据选项中显示的是什么,子串列表都会显示。同样,如果选择的对象使用输入文件数据,无论当前在示例选项中显示的是什么,子串列表也都会显示。对于其它数据源类型也一样。

对于单一条码或文字对象,子串总数及可以添加在一起的最大字符数只受可放入标签中的数据字符数目的限制。

## 添加新的数据子串到子串列表

单击子串列表下面的新建按钮 沿着子串列表选项下面的一行按钮是用于创建、编辑并重新配置多个子串。最左边的按钮是新建按钮,单击该按钮会导致右边动态数据区域发生变化,并表示正在添加新的子串。

## 重新配置子串列表

一旦建立并定义子串列表,可能会需要重新安排子串或将其删除。子串列表底部的工具栏使这些更改操作快速而容易。

每个按钮的使用方法列出如下。单击每个按钮旁边的文字,就可获得相应的信息。

**添加子串** 插入一个新的数据子串到选择对象的数据子串列表。

**删除子串** 从选择对象的子串列表中删除选定的子串。

**复制子串** 将选定的子串复制到应用程序的剪贴板中。

**粘贴子串** 将当前剪贴板中的子串粘贴到数据子串列表中。

**上移子串** 将子串列表中选定子串的位置向上移动一行。

**下移子串** 将子串列表中选定子串的位置向下移动一行。

**子串串联范例**

字符串串联的一个例子是将地址中的省份名、城市名和邮政编码字段结合成一个单一的字段。由于省份名可能比较长（如：黑龙江省），也可能比较短（如：广东省）；城市名也可能比较长（如：齐齐哈尔市），也可能比较短（如：广州市）。而省份名和城市名之间的间隔是相同的。

以下两行没有使用字符串串联功能，为了补偿长省份名和城市名，城市名字段的位置必须与省份名开始处保持相当的距离。这样就使第二行排列看上去比较难看。

```
黑龙江省齐齐哈尔市 161005
```

```
广东省 广州市      321098
```

然而，以下两行使用了字符串串联功能，城市名字段紧接着省份名字段，同时邮政编码字段紧接着城市名字段。其中第二行的显示得到了改进。

```
黑龙江省齐齐哈尔市 161005
```

```
广东省广州市 321098
```

在上述示例中，子串 1 是省份名，子串 2 是城市名，子串 3 是邮政编码。

### 结合不同的数据源

有时会发现将使用数据库类型数据与常量类型或键盘输入类型数据结合在一起生成的数据很有用。例如，假设有一个数据串，象公司名，在打印的每个标签中都会出现。一种方法是在每行数据源文件中都重复这个文本。如下显示：

```
"北京","100080","张三","京成条码公司"
```

```
"广州","510635","李四","京成条码公司"
```

```
"上海","200030","王五","京成条码公司"
```

不用在数据源文件中的每行都重复公司名，最有效的方法是在常量类型子串中指定一次。这样由于每个标签中公司名字段是不变的，在打印时不必重复地从数据库中读取。

更改后，数据库文件中的第四个字段就可以空着：

```
"北京","100080","张三",""
```

```
"广州","510635","李四",""
```

```
"上海","200030","王五",""
```

或索性去掉:

"北京","100080","张三"

"广州","510635","李四"

"上海","200030","王五"

### 子串命名和共享

数据源属性页中的变量共享名称选项用于指定子串数据是否与其它子串共享。变量共享名称选项可应用于任何数据子串，并且使用方式也相同，而不管指定的是什么数据源类型。

变量共享名称组合框用于指定当前选择的数据子串名称，以便让其它子串共享，或共享其它子串的变量。

当一个变量可能被多个对象同时使用时，应当为其指定一个共享名称，这样在其它对象中，就可以通过在变量共享名称中选中这个变量的名称，将数据连接到这个变量上。两个对象中相应的数据会同步变化。

## 使用数据工具

使用数据工具，只要单击相应的对象就可以直接修改对象数据的内容，而不能修改对象数据的属性，这个工具很适用于批量输出时更改数据。

## 第七章

# 从数据库中读取数据

## 概述

数据库是存放与特别的主题或目的相关信息的集合,比如用于打印标签或条码的信息。我们会经常从外部数据文件(例如数据库)读取数据,并用来打印条码标签。

从 LabelShop 中打印出简单标签最快捷的方式是使用数据源类型中的常量选项,直接将数据输入到标签对象中。然而,往往需要从数据源文件读取并打印其中的一些或全部数据,才能生成更复杂的标签格式,而不仅仅是从键盘输入。比如,当要输出多个标签,而且每个标签中的数据都会发生变化时,当需要使用数据库、电子表格或其它程序中的数据来打印标签时,使用分隔开的数据文件通常是首选的。

有两种数据文件类型可在 LabelShop 中引用:

**文本文件** 带有分隔符的文本文件。

**ODBC 数据源** ODBC 数据库,支持包括几乎所有主流的数据库和电子表格文件格式。

## 数据库类型

### 文本数据源

所有包含标签数据的带分隔符文本文件都需要具有一些规定的格式,以

便通知哪个字段属于标签中的对象，哪个字段属于另一个对象。这是通过使用分隔符来实现的。分隔符是一些特殊字符，用来将数据文件中的各个字段区分开。本应用程序当前支持两种类型的带分隔符的文本文件：

### 逗号分隔符

在使用逗号分隔的文本文件中，每个字段都是以逗号区分开的。以下是逗号分隔文本文件的例子：

100080,北京,北京京成条码公司,张三

510635,广州,广州京成条码公司,李四

200030,上海,上海京成条码公司,王五

由于文件中有三行文本，可以打印出三个不同的标签，每个标签有四个字段的的数据。

在每个字段中，无论是文本中间的空格，还是文本开头和结尾的空格，都会被打印出来。

### 制表符分隔符

制表符分隔符使用制表字符（TAB）区分每个字段。以下是使用制表符分隔的文件的例子：

100080 北京 北京京成条码公司 张三

510635 广州 广州京成条码公司 李四

200030 上海 上海京成条码公司 王五

由于文件中有三行文本，可以打印出三个不同的标签，每个标签有四个字段的的数据。

在每个字段中，无论是文本中间的空格，还是文本开头和结尾的空格，都会被打印出来。

默认的文本文件分隔符是逗号分隔符。这是将数据从数据库或电子表格中输出到文本文件最常用的分隔符类型。这个默认设置对大多数带分隔符的文本文件都工作得很好。然而，当数据文件字段中的数据本身包含逗号时，就需要选择另一种制表符分隔符。

### 逗号和引号混用

对于一些从电子表格引出的数据，可能在字符串的两端附加了一对引

号，使用这个选项可以同时去除字符串两侧的引号。

## ODBC 数据源

ODBC 代表“开放式数据库互连 (Open Data Base Connectivity)”。ODBC 数据源是一整套构造信息，它至少包含指定以下内容：

用于从特定数据库格式文件中读取数据的 ODBC 驱动程序。

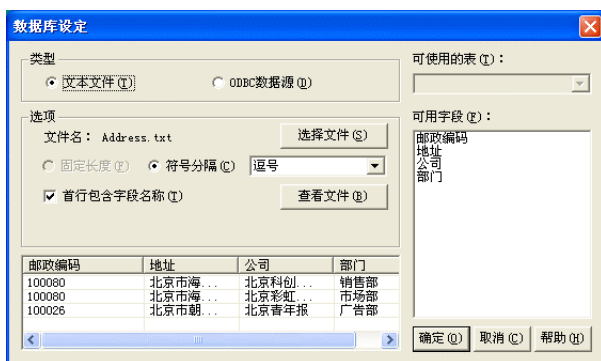
根据数据库文件格式的不同，指定目录名称或包含一个或多个数据表的文件名称。

LabelShop 具有以下能力，即从不同的数据源读取信息，而不必了解特定数据库文件格式的具体细节。这是通过使用 ODBC 驱动程序来完成的。由于可以使用很多的 ODBC 驱动程序，应用程序能够从多种数据源中读取数据。其中包括一些主要的数据库和电子表格文件格式，也包括其它标准的带分隔符的文本文件。

将位于数据库设置对话框中的类型选项设置“ODBC 数据源”，应用程序可以从带有 ODBC 驱动程序的数据源中引入数据。尽管使用已命名的 ODBC 数据源不需要更多关于 ODBC 或 ODBC 管理器的知识，但了解一些 ODBC 的基本知识会使工作更容易。应该注意，从非文本类型的文件中读取数据而不使用 ODBC 管理器几乎是不可能的。

## 数据库对话框

数据库设置对话框用于设置显示或打印时读取数据的数据库。应用程序将从数据库读取记录，并将原始数据传入标签上经过配置的对象中。



数据库设置对话框中的类型选项用于设置以下数据库选择：

**文本文件** 支持带分隔符的文本文件。

**ODBC 数据源** 支持几乎所有数据库文件格式，包括作为使用最多的电子表格。

## 数据库选项

### 文本文件选项

**文件名** 用于显示当前选定的文本文件名。

**选择文件** 用于显示打开对话框，指定要引入的文本文件。

**查看文件** 用于调用系统默认文本编辑器，查看当前选定的文本文件。

**符号分隔** 用于指定文本文件中区分每个字段的分隔符。

**首行包含字段名称** 将文件的第一行解释为字段名。

### ODBC 数据源选项

**数据源名** 用于显示数据源名称。

**选择数据库** 用于显示选择数据源对话框，指定要引入的数据源名称。

**库类型** 用于显示数据库的类型。

**库文件** 用于显示当前选定的数据库文件名。

### 其它选项

**可使用的表** 用于选择当前数据库文件中可用的数据表。

**可用字段** 用于选择当前数据表中可用的字段。

**数据显示** 用于显示当前数据表中的样本数据。

---

**注意：**对于不同的 ODBC 数据源选项，可用的表的名称所代表的内容可能不同。对于数据库而言，表的名称包括了 TABLE 和 VIEW，对于 Microsoft Excel 电子表格数据，表的名称是 SHEET 的名称，对于 DBASE，表的名称是 DBF 文件的名称。

如果打开含有数据库联接的标签格式文档时，数据库不能被正确联接，则所有的数据库类型变量的值会被一个空字符串替代。在联接文本数据源时，空行也被作为一条所有字段都为空的记录进行处理。

可以在任何时候通过“数据库——删除数据库”菜单命令移去数据库联接。

---

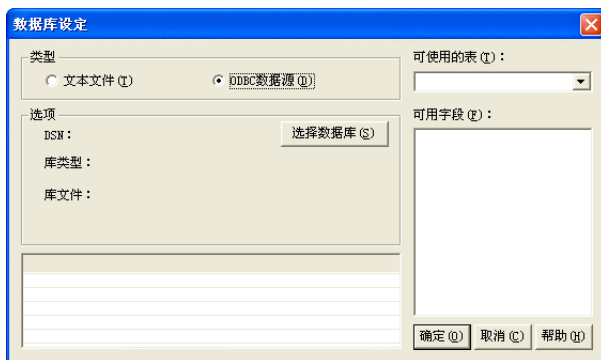


## ODBC 数据源连接示例

### dBASE 及 FOXPRO 文件:

dBASE 及 FOXPRO 文件版本较多，此处以一个 dBASE III 文件为例说明连接过程：

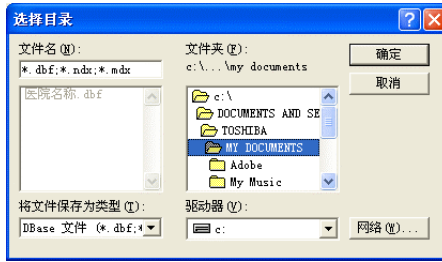
1、首先在数据库对话框中选中“ODBC 数据源”，如下图



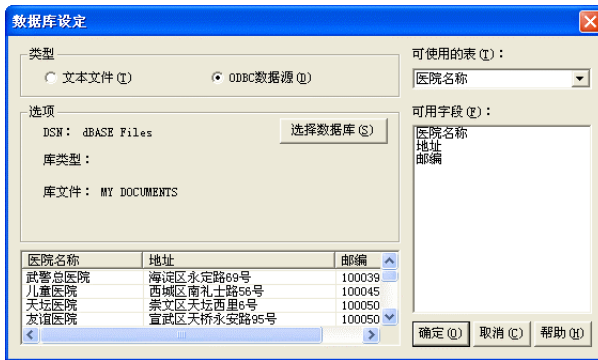
2、点击“选择数据库”，在“选定数据源”对话框中选择“机器数据源”，选中 dBASE Files，点击确定；



3、在文件夹中选中 DBF 所在的目录，点击确定；

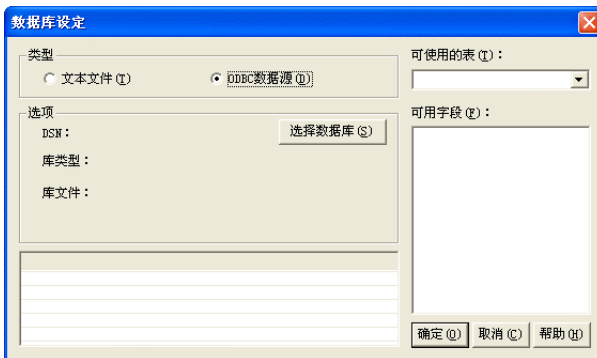


4、所选定的目录中的所有 dBASE 文件会列出在可使用的表中。



## Excel 电子表格：

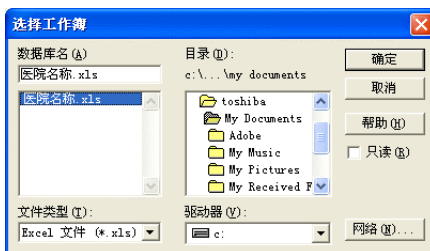
1、首先在数据库对话框中选中“ODBC 数据源”，如下图



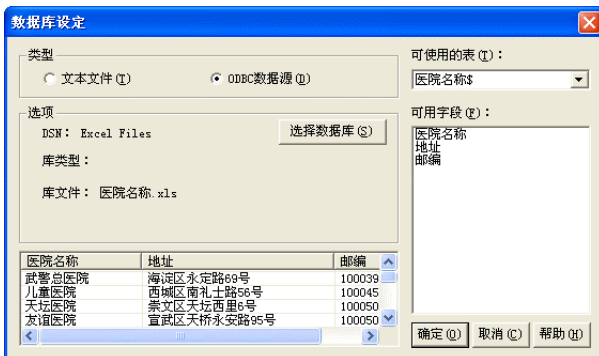
- 2、点击“选择数据库”，在“选定数据源”对话框中选择“机器数据源”，选中 Excel Files，点击确定；



- 3、在对话框中选中要连接的电子表格文件，并点击确定；

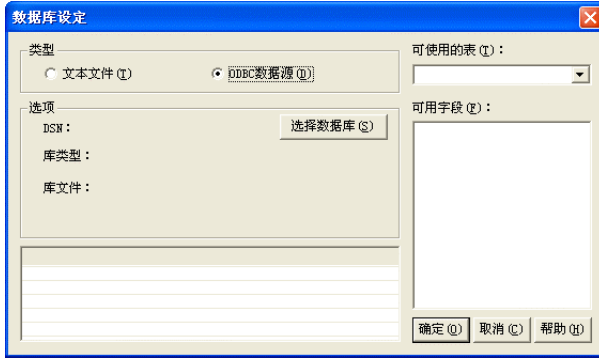


- 4、如下图所示，Excel 文件已连接成功。

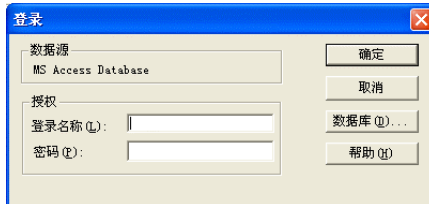
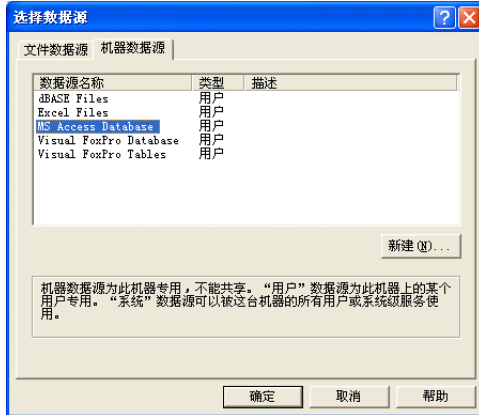


### MS Access 文件:

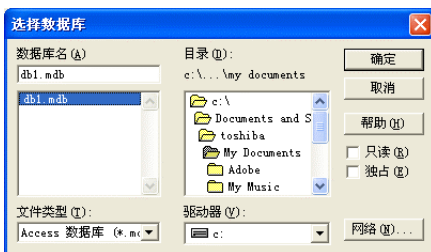
- 1、首先在数据库对话框中选中“ODBC 数据源”，如下图



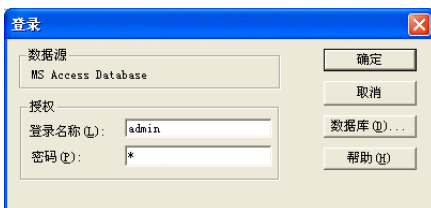
- 3、选择数据库，在“机器数据源”，选中 Ms Access Database，如果需要，输入登录信息；



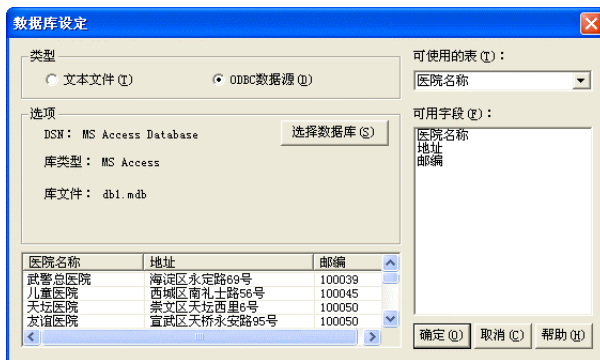
## 3、选中需要连接的 Access 文件；



## 4、如果需要，再次输入登录信息；



## 5、如下图所示，从可用表中选中要使用的表，数据库连接成功。



## 附加说明

在使用 ODBC 型数据源时，不建议使用文件数据源。



## 第八章

# 打印输出标签格式

## 概述

打印输出满足要求的标签是用户使用 LabelShop 的最终目的。一旦标签格式显示在屏幕标签查看区域,并符合打印输出的需要,就可以使用打印标签对话框来打印输出标签。

---

**重要信息:** 对于标准的图形输出方式,全部位于标签外面的对象是不被输出的,而对于命令输出方式,只有完全位于标签范围内的对象才能打印出来。另外,对象的不打印输出属性和系统的输出非打印对象设置也会决定某些对象是否被输出。

---

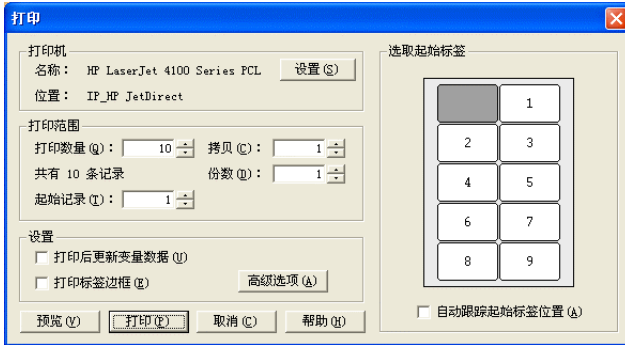
在不同的打印机上,同一个标签的输出结果可能不同,这是由于不同的打印机可能具有不同的输出精度等属性,而标签格式中很多对象的参数可能依赖于这些属性。

另外,对于专用的条码标签打印机采用命令方式输出时,打印机的专用字体可能显示与输出结果存在一定的差异,应以实际输出结果为准。在标准的图形输出方式下,LabelShop 支持打印预览功能,建议在打印输出标签前使用该功能检验一下将要输出的内容,这样可以避免不必要的浪费。

对于特定的打印机,有可能需调整边距以便使标签的输出位置与实际的标签纸对准,这可能需要经过几次输出调整才能最终完成。

## 打印对话框

打印对话框用于指定如何打印标签。打印对话框中的选项用于指定标签打印份数和序列标签拷贝数，还有一些附加的打印选项。对于标准图形输出方式和命令输出方式，可供选择的项目会有所不同。



### 打印机

这部分用于显示打印标签所使用的打印机的有关信息。

**名称** 显示指定用来打印标签的打印机名，如果而要更改目标打印机，请在标签设置对话框中进行。

**位置** 显示打印机安装的端口位置或网络路径。

**设置** 设置打印机的属性。为方便起见，打印机的设置都存储在标签格式式文档中。这样，下次在需要打印相同的标签时，要作的只是装入格式式文档并打印。

在开始设计一个标签格式式文档之前，首先需要选择一个合适的打印机，以便标签格式按照正确的打印机规格设置。这样做对于标签中包含条码对象的情况就格外重要。对于激光打印机和针式打印机来说，不同的条码密度范围可能会影响给定标签的尺寸。比如，当选择了一个激光打印机，放置在标签中的条码就可能比选择一个针式打印机时要短。

标签格式式中可保存打印机的一些特殊选项，在下次打印输出时，会用同样的设置来配置打印机，而不论打印机的默认设置如何。同样，标签格式式中也记住了对应的打印机，标签只在相应的打印机上输出，而不论该打印机是否是系统默认的打印机。



## 打印范围

这部分内容用于指定打印标签的数量。

**打印数量** 打印对话框中的打印数量选项用于指定打印标签的数量。例如，如果要打印二十个标签，只要在单击打印按钮之前在打印数量编辑框输入 20。实际输出的标签个数还受拷贝数和份数影响，此处的打印数量会对序列号变量和数据库类型变量产生影响。当有数据库连接时，打印数量指定了输出记录的个数，序列号变量也按此数量进行序列化。

**拷贝数** 如果相同的标签需要打印多份，则需要指定拷贝选项。

**份数** 对于页式打印机，如激光打印机，如果相同的标签需要打印多页，则需要指定份数选项。

---

**极为重要的说明：**打印数量、拷贝数、份数三者有很大的区别。

打印数量一般是指打印变化标签的数量。当标签格式中全部对象的变量都是常量时，此项与拷贝数作用相同，而当变量的类型是序列号或数据库等时，打印数量指定的是变化的标签的数量。

拷贝数是指同一个标签的重复数目，当标签中对象的变量是序列号等类型时，拷贝数一般指定的是同一个标签需要重复输出的数目。

份数只有在页式打印机上才有意义。

---

## 设置

**打印后更新变量数据** 用于设置完成打印作业后，是否自动更新变量数据，如序列号。

当打印对话框中的打印后自动更新数据变量选项选中后，序列号子串数据在完成打印作业后，将增加或减少后的数据写回到序列号数据中去，以便下次打印时，从更新的数据开始。

**打印标签边框** 用于指定打印标签内容时，是否将标签的边框线一起打印出来。此选项对于命令输出方式无效。

## 选择起始标签

当打印一张标签时，通常是从左上角开始打印。然而，有时需要指定从哪个标签开始打印，这样做是为了能在一张有部分标签已经使用过的标签上继续打印。

在打印对话框的右边，用鼠标单击要开始打印的标签。被指定为起始标签之前的标签都将变成灰色，而之后的所有标签将重新排号，指定的起

始标签排号为 1。

当一页标签纸上的标签不能一次全部输出时，通过选定起始标签位置，可将此标签纸上未输出和标签进行输出。但由于标签纸的特殊性，一般不建议在同一张标签纸上多次打印。

**自动跟踪起始标签位置** 选项用于设置每次执行打印任务后，系统将根据标签的实际输出数量自动更新标签的起始位置。

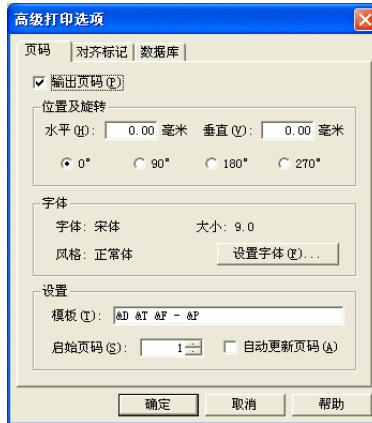
对于命令输出方式不起选择起始标签位置。

## 高级打印选项

用于设置有关页码、对齐标记和标签拷贝份数等信息。

### 页码属性页

在标准图形输出方式下设置页码信息。当同时打印多页标签时，可能需要对每页标签纸标注页码，以方便对标签进行使用和管理。LabelShop 提供了可在每张标签纸上输出页码的能力，用户可对页码的内容、字体、位置等进行设置。



**输出页码** 指定是否在每一页标签纸上输出页码信息。

**位置及旋转** 设定页码的位置及角度

**字体** 设定页码文字的字体

**模板** 设定如何格式化页码信息。其中 &D 代表当前日期，&T 代表当

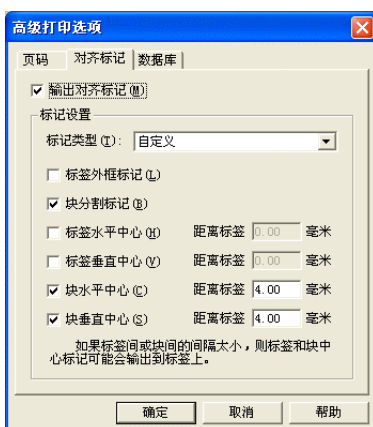
前时间，&F 代表当前标签格式文档名称，&P 代表当前页码。

**启始页码** 设定当前打印任务的启始页码号

**自动更新页码** 设定是否在打印输出完成后自动更新页码。

## 对齐标记属性页

当输出是在非标签纸上时，有时可能需要输出标签的边界位置作为标签裁切的对位标记，在一些特殊场合（如证卡的制作过程中），还需要输出一些特殊的对位标记，LabelShop 为用户提供了这些附加功能。



**输出对齐标记** 选项用于设置是否输出对齐标记。

**标记类型** 用于选择输出何种标记：

**标签外框标记** 标出标签的四个角的位置。

**块分割标记** 用于标出块与块之间的位置，用于将标签块分开。

**标签水平中心** 标记出每个标签的水平中心位置。

**标签垂直中心** 标记出每个标签的垂直中心位置。

**块水平中心** 标记出每个标签块的水平中心位置。

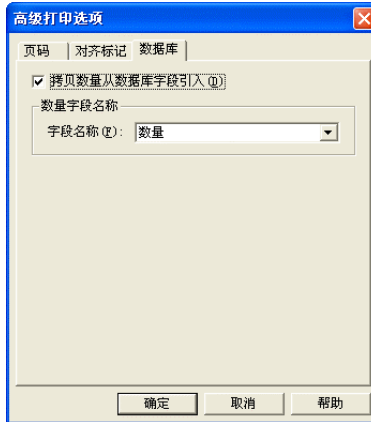
**块垂直中心** 标记出每个标签块的垂直中心位置。

**距离标签** 用于设置标记与标签或标签块边界的距离。

## 数据库属性页

当标签格式中含有数据库联接时,有时可能会有一个数据表字段记录着标签需要输出的拷贝数。LabelShop 中可指定从数据表字段中提取标签的拷贝数量。

**拷贝数量从数据库字段引入** 选项指定是否由一个数据库字段决定每个标签的拷贝数。



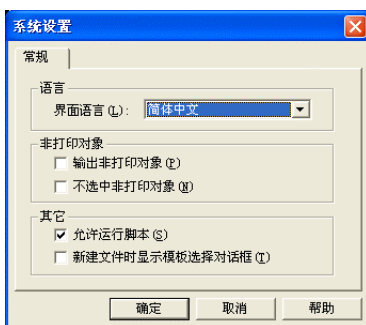
**字段名称** 用于选择引入拷贝数量的字段。

如果指定了从数据库字段引入标签输出拷贝数量,则打印对话框中的拷贝数量输入框将无效。

## 第九章

# 系统参数设置

系统选项对话框用于设置 LabelShop 的系统参数。



**界面语言** 用于设置 LabelShop 界面的语言，目前有简体中文、繁体中文和英文可供选择。对于繁体中文和英文，而要相应的 DLL 文件支持，且 DLL 的版本要与 LabelShop 的版本相同。

可以从 LabelShop 的网站下载到最新版本的程序和 DLL。

**输出非打印对象** 用于设置可以输出具有非打印属性的对象。

**不选中非打印对象** 用于设置在设计标签格式时所有具有非打印输出属性的对象不能被选中，仅作为背景显示。如果想编辑这些对象，请清除此选项。

**允许执行脚本** 设置是否可以执行脚本变量中的 VBScript，使用脚本变量可以实现一些高级数据处理功能。

**新建文件时显示模板选择对话框** 设置在新建标签格式时，是否可以从预先保存的标签模板中进行选取。要实现此功能，需要在 LabelShop 所在的目录中建立 Template 子目录，并且将模板文件保存到此目录中。

## 第十章

# 菜单命令参考

下面详细介绍 LabelShop 的菜单功能。

## 文件(&F)菜单

新建(&N)：新建一个 LabelShop 文档。

打开(&O)...：打开一个保存的 LabelShop 文档或模板。

关闭(&C)：关闭当前编辑的文档。

关闭全部文件(&F)：关闭当前打开的全部文档。

保存(&S)：保存当前编辑的 LabelShop 文档。

另存为(&A)...：换名保存当前编辑的 LabelShop 文档。

另存为模板(&T)...：将当前编辑的 LabelShop 文档另存为模板。

打印(&P)...：打印当前编辑的文档。

打印预览(&V)：预览当前编辑的文档。

标签设置(&L)...：设置当前文档的标签设置。

说明(&M)：设置当前文档的附加说明。

输入注册码(&R)...：输入 LabelShop 的注册码。

**最近文件：**快速打开曾经编辑过的 LabelShop 文档。

**退出 (&X)：**退出 LabelShop 程序。

## 编辑(&E)菜单

**撤消 (&U)：**撤消上一步操作。

**恢复 (&R)：**恢复刚刚撤消的操作。

**剪切 (&T)：**将当前选取的全部对象删除并复制到剪切板中

**复制 (&C)：**将当前选取有全部对象复制到剪切板中。

**粘贴 (&P)：**将剪贴板中的对象放置到标签查看区域中。

**全选 (&A)：**将全部对象全部选取。

**删除 (&D)：**将当前选取的对象全部删除。

**属性：**显示当前选取对象的属性设置对话框。

**插入新对象 (&N)...**：插入 OLE 对象

## 查看(&V)

**工具栏 (&T)：**显示或隐藏工具栏。

**格式栏 (&F)：**显示或隐藏格式栏。

**对齐栏 (&A)：**显示或隐藏对齐栏。

**状态栏 (&S)：**显示或隐藏状态栏

**数据库栏 (&D)：**显示或隐藏数据库栏。

**显示对象说明 (&R)：**是否显示选中的对象的附加说明。

**撑满窗口 (&W)：**将标签撑满窗口显示。

**放大 (&I)：**放大标签的显示比例。

**缩小 (&O)：**缩小标签的显示比例。



## 工具(&T)菜单

- 选取(&S)：选择选取工具。
- 条码(&B)：选择新建条码对象工具。
- 线条(&L)：选择新建线条对象工具。
- 图片(&P)：选择新建图片对象工具。
- 矩形(&R)：选择新建矩形对象工具。
- 文字(&T)：选择新建文字对象工具。
- 数据(&D)：选择修改对象数据工具。
- 放大(&I)：选择放大显示比例工具。

## 排列(&A)菜单

- 组合(&G)：将所选取的对象组合成一组。
- 取消组合(&U)：取消选取的对象中的组合对象的组合。

### 对齐菜单：

- 左齐：将所有被选取的对象左侧对齐。
- 右齐：将所有被选取的对象右侧对齐。
- 顶齐：将所有被选取的对象顶部对齐。
- 底齐：将所有被选取的对象底部对齐。
- 垂直中齐：将所有被选取的对象垂直中间对齐。
- 水平中齐：将所有被选取的对象水平中间对齐。
- 水平居中：将所有被选取的对象放置在标签水平中心。
- 垂直居中：将所有被选取的对象放置在标签垂直中心。

### 尺寸菜单：

- 水平同宽：将所有被选取的对象水平方向尺寸设定与参考对象的宽度相同。

**垂直同宽：**将所有被选取的对象垂直方向尺寸设定与参考对象的高度相同。

**水平垂直相同：**将所有被选取的对象水平、垂直方向尺寸设定与参考对象相同。

**间距菜单：**

**水平间距相同：**将所有被选取的对象水平方向均匀分布。

**垂直间距相同：**将所有被选取的对象垂直方向均匀分布。

**旋转菜单：**

**左旋 90 度：**将所有被选取的对象逆时针旋转 90°

**旋转 180 度：**将所有被选取的对象旋转 180°

**右旋 90 度：**将所有被选取的对象顺时针旋转 90°

**位置锁定/取消位置锁定(&L)：**设置或取消被选取对象的锁定属性。

**移到最前：**将被选取的对象上移到最前面。

**前移：**将被选取的对象上移一层。

**后移：**将被选取的对象下移一层。

**移到最后：**将被选取的对象下移到最后面。

**数据库(&D)菜单：**

**设置数据库(&D)...**：设置文档的数据库属性。

**定位记录(&S)：**定位数据库到一个新的记录。

**更新数据库：**更新数据库数据。

**第一条记录：**移到第一条记录。

**上一条记录：**移动到前一条记录。

**下一条记录：**移动到后一条记录。

**最后一条记录：**移到最后一条记录。

**删除数据库(&E)：**从文档中移去数据库联接。

**选项(&O)菜单:**

系统选项 (&C) . . . : 设置 LabelShop 的系统参数。

**窗口(&W)**

新建窗口 (&N) : 为当前打开文档创建一个新的窗口。

层叠 (&C) : 堆叠并排列所有打开的窗口, 可以查看到每个窗口的标题栏。

横向平铺 (&H) : 排列所有打开的窗口, 使各个窗口在标签查看区域一个堆叠在另一个的下方。

纵向平铺 (&V) : 排列所有打开的窗口, 使各个窗口在标签查看区域一个并排放在另一个的旁边。

排列图标 (&A) : 将所有最小化的窗口排列在 LabelShop 标签查看区域的下方。

**帮助(&H)菜单:**

帮助主题 (&H) : 访问 LabelShop 帮助系统的帮助主题。

关于 LabelShop (&A) . . . : 显示关于 LabelShop 这个版本的信息。



附录 A

# 如何做

此处介绍一些常用功能的实现方法。

## 1、如何快速更改标签显示比例：

在 LabelShop 底部状态栏在左侧有一个显示比例输入框，在其中直接输入显示比例并回车，就可快速更改显示比例。

## 2、如何使标签完全显示在窗口中：

使用撑满窗口工具。

## 3、如何预览多个标签：

执行打印命令，在打印对话框中输入需要预览的标签数目，再执行预览命令。

## 4、如何从一页有左下角开始输出标签：

改变标签设置对话框中其它属性页中的起始位置为“左下角”即可。

## 5、标签纸上面已打印了一部分标签，如何继续使用：

您可在打印对话框中选择第一个标签的输出位置。

## 6、如何使已排好的对象不被移动：

选取该对象，通过设置锁定属性锁定该对象的位置。

**7、如何为标签设置一个不输出的参考背景：**

可将背景对象设置为具有“不打印输出”属性的对象。

**8、如何使一个对象不被打印输出：**

通过设置该对象的“不打印输出”选项实现。

**9、如何在编辑标签格式时不选中非打印对象：**

通过设置程序选项中的“不选中非打印对象实现”。

**10、如何打印号码顺序变化的标签：**

可以使用“序列号”类型的变量。

**11、如何将多个对象的数据设定为同一个数据：**

使用共享变量。

**12、如何提高专用条码打印机的输出速度：**

选择并口的打印机，尽量使用打印机字体，使用 LabelShop 安装盘上的打印机驱动程序。

**13、如何为每个标签输出不同的图片：**

使用数据库和具有“数据源图片”属性的图片。

**14、如何输入 EAN128 条码的 FNC1 字符：**

选中“EAN128”条码，在数据中输入“^1”即为 FNC1 字符。

**15、如何实现一些复杂的数据格式化处理功能：**

可以在通过脚本数据类型中编写 VBScript 实现。

**16、如何实现 LabelShop 中未包括的校验算法：**

使用 VBScript 功能。

**17、如何让 VBScript 运行起来：**

LabelShop 默认不执行 VB 脚本，需要在系统选项中进行设置允许执行脚本。

## 18、如何中止并修改错误的脚本：

当脚本进入死循环时，需要强制中止 LabelShop，并在系统设置中禁止脚本执行，然后再打开出错的标签格式，修改出错的脚本。

## 19、如何在输出时快速更改数据：

可以使用“数据”工具。

## 20、如何实现证卡的双面位置对齐：

可以将证卡的反面的标签起始位置设定为右上角，同时使用对齐标记。

## 21、如何快速画出一些间距相等的线：

首先画出全部线条，然后确定第一条和最后一条线的位置，再使用水平或垂直间距相同命令。

## 22、如何建立一个新的标签设置：

在新建标签时随意选择一个标签设置，然后在此基础上进行修改。

## 23、如何使邮政编码精确打印在信封的邮编框中：

需要设置字间距。

## 24、如何将图标保存在标签格式中：

使用嵌入图片，将图片数据放入标签格式文件中。

## 25、如何在网络中让多个用户同时使用 LabelShop：

您需要购买网络版本的 LabelShop。

## 26、如何将版面功分成多个区域：

可以使用标签分块功能。

## 27、如何输出长字或扁字：

设置字体属性中的“字体宽度方向缩放倍数”。

## 28、如何实现字体的反相输出：

设置字体属性中的“黑底白字”。

### **29、如何更改条码的大小：**

更改条码的 X 方向尺寸。

### **30、如何在数据输出时在数据中插入字符（如空格）：**

使用字符模板。

### **31、如何使标签内容与模切的标签纸对齐：**

设置标签格式的位置微调或标签格式的左边空和上边空。

### **32、如何快速熟悉 LabelShop 的功能：**

请参考 LabelShop 附带的示例标签格式，LabelShop 的示例文件被安装在 LabelShop 应用程序目录中的 Samples 目录中。

### **33、如何获得更多的帮助：**

您可以通过电话或电子邮件与软件经销商联系，或者直接与软件开发商——北京科创京成条码科技有限公司联系。



## 附录 B

# 为什么

## 1、为什么 EAN、UPC 条形码的标注显示不美观？

当运行 LabelShop 时，出现在 EAN 或 UPC 等条码对象标注中的数字和字符不美观，有重叠的现象，这是由于您的系统中没有 OCR-B 字体。一般在安装 LabelShop 时会自动所带的 OCR-B 字体文件。

解决方法是安装 LabelShop 所带的这两种字体文件。也就是在安装软件时选择“完全安装”或在“自定义安装”中选择“字库”选项。

## 2、标签中图像为什么显示不出来？

标签中的图像曾经能够显示，但现在却显示不出来了。出现这种情况的多数原因是：

- 1) 图像对象属性中图象文件名指定的路径不正确，或原来没有指定路径名，但当前系统所在的目录与原图像默认的路径名不一致。
- 2) 所指定路径下没有该图像文件或图像文件已被删除。

解决方法：

- 1) 可以重新设置图像文件的路径名。
- 2) 将图像复制到指定路径下。
- 3) 在图片属性中选择“嵌入式图片”。

对于新引入的图片，可能是该图片的格式不被 LabelShop 支持（例如 CMYK 的图片）。

解决办法：

在 Photoshop 等图像处理软件中重新将图片另存为 LabelShop 支持的格式（如 BMP 或 RGB 的 TIF 格式）。

### **3、当运行 LabelShop 时，显示信息对话框“未安装加密狗或加密狗不正确！”**

这是由于没有将系统所带的加密狗插到计算机的并行端口上，也可能是由于尽管加密狗插在了并口上但并没有插紧。请退出 Windows 后关闭计算机，并将加密狗插在并口或将其插紧。重新开机并再次运行 LabelShop，如果仍然出现这种现象，就很可能是加密狗中的系统参数丢失，这就需要和软件销售商联系。

### **4、为什么有些对象无法移动？**

在运行 LabelShop 时，移动选定的对象会出现有些对象发生了移动，而有些对象却无法移动，出现这种情况的原因是由于每个对象都有一个特征，即位置锁定。

如果要更改这一特征，用鼠标左键双击对象，弹出属性对话框，在“常规”项目中，修改位置锁定属性。

### **5、为什么输入注册码无效？**

软件和加密狗是一一对应关系，在用户购买软件时，会得到一个唯一的注册号，它是与加密狗相对应的，当软件安装完成后，运行 LabelShop，系统会提示输入注册号。

如果软件的注册号和加密狗不对应，则输入的注册无效。请与您的软件经销商联系。