

文章编号: 1005-8451 (2004) 02-0037-03

## 在 Windows NT 和 Windows 2000 下自定义打印

杨帆

(乌鲁木齐铁路局 电子计算中心, 乌鲁木齐 830011)

**摘要:** 运用 Windows API 函数和 VB 的基本编程技巧与方法, 提出了一种在 Windows NT 和 Windows 2000 操作系统下自定义纸张打印类的设计和实现方法。并可以在其它语言如 PB 中调用该类库。

**关键词:** Windows 2000; 自定义纸张; 打印; VB; PB

在 Windows 95、Windows 98 和 Windows Millennium Edition(Me) 操作系统中可灵活地自定义页面大小。但在运行 Windows NT、Windows 2000 和 Windows XP 的系统上, 用户在使用前必须先将所有页面大小定义为表单 (Form) 才能使用。本文主要介绍在 Windows NT 和 Windows 2000 下用户自定义纸张的打印, 并附加程序代码来说明如何增加、选择用户定义的表单 (Form)。

### 1 基本原理

在运行 Windows 95、Windows 98 和 Windows Millennium Edition(Me) 的系统上, 程序员可以简单地把 DEVMODE 结构中的 4 个成员设置为: dmPaperSize = 256 (纸张类型); dmFields = DM\_PAPERWIDTH + DM\_PAPERLENGTH; dmPaperLength = (指定的纸张长度或高度); dmPaperWidth = (指定的纸张宽度)。

再通过 API 函数调用 DocumentProperties, 就可以达到使用自定义纸张、随意改变纸张大小的目的了。

微软公司为方便 Windows 操作系统的普通用户 (非程序员) 考虑, 从 Windows NT 开始 (包括 Windows XP 和 Windows 2000), 不再支持这种简单的自定义纸张使用方法, 而是改为使用用户已经预定义好的表单 (Form, 也称“页型”, 这可通过打印机的“服务器属性”进行调整) 来确定纸张大小。此时, 程序员只能通过设置 DEVMODE 结构中的 dmFormName (表单名称) 或者对 dmPaperSize (纸张类型) 赋值的方法来选择不同的纸张, 不允许直接改变 DEVMODE 结构中的 dmPaperLength (纸张长度或高度) 和 dmPaperWidth (纸张宽度)。若程序中需要使用特殊

大小的纸张类型, 程序员或用户就必须预先定义这样一个指定大小的表单。

在微软知识库中, 就是按这个思路提出了一个解决该问题的方法, 即: 通过使用 EnumForms 函数枚举当前打印机支持的所有表单 (Form)、并检查每个表单的高度和宽度, 如果找到了指定大小的表单, 则选择它; 如果未找到指定大小的表单, 则使用 AddForm 函数添加该表单、并选择它。

但是, 如果象铁路财务软件系统那样, 由于自定义纸张类型很多、并且其中的许多打印输出允许用户任意指定纸张大小, 这就可能存在问题是。如果存在太多的用户自定义纸张, 不便于用户记忆。本文提出了另外一种有效的处理方法来妥善解决该问题:

使用一个特定的表单 (Form), 来代替 Windows 98 下原来的打印机纸张类型 256 (自定义纸张), 其它的表单都假定是系统预设的, 不允许调整大小 (Size)。当检测到该表单不存在时, 自动增加 1 个表单; 如果当前要求的表单大小与该表单的当前设置不一致, 则首先修改该表单设置、并选择之。此外, 为了适应不同的语言环境应用需要, 在具体实现时要求实际修改打印机的当前设置;

本文的处理办法与微软所建议的完全不同。微软建议: “检索打印机的 DEVMODE 结构, 设置 dmFormName 成员, 并使用 ResetDC 函数将此表单 (Form) 设置为打印机设备上下文 (DC) 的当前表单 (Form)。选择此表单 (Form) 作为 DC 的当前表单 (Form) 时, 它仅对调用进程有效, 而不会更改驱动程序的默认设置。这不需要任何特殊权限。对于需要更改打印机设置的应用程序, 建议使用此方法”。如果应用程序使用固定的打印机设备上下文 (DC) 的话 (如 VB), 微软所建议的当然是完全正确的。并不是所有的应用程序都使用固定的打印机设备上下文 (DC), 例如允许使用不同打印作业 (Print Job) 的

收稿日期: 2003-07-30

作者简介: 杨帆, 工程师。

PowerBuilder, 其打印机设备上下文 (DC) 并不是一成不变的。为了适应这种情况, 本文对系统用户的权限提出了更高的要求: 要求用户对需要其输出内容的打印设备拥有完全的控制权限, 不论该打印设备是本地打印机或网络打印服务器。

## 2 具体实现

### 2.1 基本处理流程

(1) 建立 1 个标准 EXE 工程, 在工程中添加一个模块, 在模块中声明 API 函数及相关常量, 并建立自定义函数:

a. ChangeDefaultPrinter ()。功能描述: 将指定的打印机设定为系统缺省打印机, 修改注册表中打印机名称和打印端口号, 并取得该打印机所支持的最大、最小纸张尺寸。

b. GetDefaultPrinter ()。功能描述: 取得系统缺省的打印机名称。

c. GetPrinterPaperList ()。功能描述: 取得指定打印机所有可用的纸张类型的信息, 即: 纸张名称、尺寸、编号, 并能自动增加或调整自定义纸张。

d. GetPrintersList ()。功能描述: 取得系统所安装的打印机的列表。

e. SelectForm ()。功能的描述: 根据表单名选择表单。

判断如果表单名等于指定的表单名称, 则调用根据纸张号选择表单的函数。

f. SelectFormByPaperNo ()。功能描述: 根据纸张号选择表单。

在该函数中首先检查是否为合法用户并初始化打印机纸张设置, 同时由于要调用 API 函数 SetPrinter, 所以如果在 Windows 2000 系统下就要求设置打印机的访问权限 PRINTER\_ALL\_ACCESS。其次要判断系统是否支持用户自定义纸张; 如果当前是自定义纸张, 则根据表单纸张的大小改变打印机纸张设置并调整打印方向、打印质量、文档份数、色彩、纸张源、比例等项的值。

g. 类的属性 GetUserDefinitionFormName ()、 GetUserDefinitionPaperNo ()。功能描述: 得到用户自定义的表单名称和纸张号。

(2) 调试编译为 Win2kPrinterClass.dll。

流程见图 1。

### 2.2 特殊问题的处理

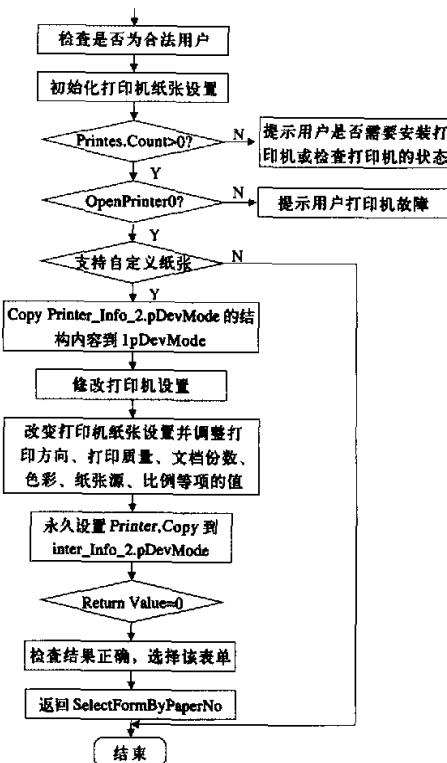


图 1 基本处理流程

a. Windows 中的 API 调用: DeviceCapabilities, 其在 Windows 2000 下返回的打印机所支持的最大/最小纸张 (DC\_MINEXTENT、DC\_MAXEXTENT) 并不正确。为此, 具体实现时采取补救措施, 新增加函数 PaperSizeExtent() 用于取得打印机支持的最合适的纸张大小 (即: 纸张的最小/最大宽度、最小/最大长度)。

b. PB7 可以调用该打印类库, 但由于 PB7 对 DBCS 支持不好, 所以所有涉及字符串调用的地方, 都是以 Byte() 返回的。

### 3 在 PB 中调用该打印类

(1) 首先, 在合适的地方添加一变量定义 (可以是全局变量):

OLEObject MyPrinter

(2) 在合适的地方实例化该打印类 (如框架的 MDI 窗口的 Open 事件):

If Not IsValid(MyPrinter) Then MyPrinter = Create

## OLEObject

```
MyPrinter.ConnectToNewObject('Win2kPrinter.
Win2kPrinterClass')
(3) 在合适的地方销毁该打印类实例 (如框架的
MDI 窗口的 Close 事件):
If IsValid(MyPrinter) Then Destroy MyPrinter
```

(4) 在需要使用自定义纸张的地方, 调用该打印类的 SelectForm 方法:

函数定义是: Function string SelectForm(String FormName,

```
Ref Long Form_cx, Ref Long Form_cy,
Ref Boolean ChangeFormSize, Long hWnd,
Optional Long Orientation = 0, Optional Long Zoom = 0,
Optional Long PrintQuality = 0, Optional Long
Copies,
Optional Long ColorMode = 0, Optional Long PaperBin)
```

// 表单名称(缺省为“铁路财务 4.0 自定义纸张”),  
纸张宽(0.1mm)、纸张高(0.1mm)、  
是否改变纸张大小、窗口句柄、  
打印方向(1 纵向 2 横向)、  
缩放比例(缺省 100)、  
打印质量(-1 草稿打印质量、-2 低级打印质量、-3  
中等打印质量、-4 高级打印质量)、  
打印份数、  
颜色模式(1 单色, 2 彩色)。

进纸纸盒(1 从上层纸盒进纸, 2 从下层纸盒进纸  
3 从中间纸盒进纸 4 等待手动插入每页纸, 5 从信封  
进纸器进纸, 6 从信封进纸器进纸, 但要等待手动  
插入, 7 (缺省) 从当前缺省纸盒进纸, 8 从拖拉进  
纸器供纸, 9 从小型进纸器进纸, 10 从大型纸盒进  
纸, 11 从大容量进纸器进纸 14 从附加的卡式纸盒  
进纸)

## (5) 其它的类属性、方法

// 也可以通过纸张号选择纸张 (与 SelectForm 基  
本相同, 其中 PaperNoID 可以通过 GetPrinterPaperList()  
取得)

Function string SelectFormByPaperNo(Ref Long  
PaperNoID,

```
Ref Long Form_cx, Ref Long Form_cy,
Ref Boolean ChangeFormSize, Long hWnd,
Optional Long Orientation = 0, Optional Long Zoom = 0,
Optional Long PrintQuality = 0, Optional Long Copies,
Optional Long ColorMode = 0, Optional Long PaperBin)
```

// 取得打印机列表: 返回值\_\_(空)失败、否则成  
功

Function String GetPrintersList()

// 取得缺省打印机 (打印设备名称): 返回值\_\_  
(空)失败、否则 成功

Function String GetDefaultPrinter()

// 改变缺省打印机 (打印设备名称): 返回值\_\_  
True 成功、False 失败

Function Boolean ChangeDefaultPrinter(String  
PrinterName)

// 设置打印机自定义纸张表单: 只有当该表单  
名称与 SelectForm 方法中的表单名称一致时, 才允许  
自定义纸张

// Property Let WorkFormName(String FormName)

// 取得当前的打印机自定义纸张名称

Property Get String GetUserDefinitionFormName()

// 取得当前的打印机自定义纸张号

Property Get String GetUserDefinitionPaperNo()

// 取得当前打印机支持的纸张(Tab 分隔)

Function String GetPrinterPaperList()

(6) 其它

使用前, 需要注册 Win2kPrinterClass.dll (可通过  
安装程序处理); 由于 PB7 目前不能正确处理 dll 函数  
返回的 DBCS 字符串, 需要对返回结果为 string 的 dll  
函数使用 FromUnicode 和 String 函数进行“字符集转  
换”处理; 对网络打印机 (非本地口的网络打印机)  
的支持与用户的权限有关。

## 4 结束语

本文提出了在 Windows NT 和 Windows 2000 下用  
户自定义纸张打印的新方法。目前该打印类库已  
经应用于铁道部铁路财务软件, 在 PB 语言环境下调用  
并实现自定义纸张的打印。

## 参考文献:

- [1] Application Visual Basic6.0 Win32 API 开发人员指南 [M]. 京  
京翻译组. 北京: 机械工业出版社, 2000.
- [2] Simon J.A. PowerBuilder7.0 使用全书 [M] (第 3 版). 北京: 电  
子工业出版社, 2002.
- [3] 于 刚. 热转印打印机的工作原理与维护 [J]. 铁路计算机应  
用, 2002, 12 (1).