

C++打印机首选项设置

原创

Li_Zhi_Yao

于 2020-10-15 18:28:01 发布

阅读量5.4k

收藏 20

点赞数 6

版权


分类专栏：

C/C++

打印机

文章标签：

c++

 C/C++ 同时被 2 个专栏收录 ▾

2 订阅 18 篇文章

订阅专栏

摘要 本文档介绍了如何使用C++通过Windows API控制打印机的首选项设置，特别是针对彩色和黑白打印的选择。首先检查打印机是否支持颜色设置，然后通过`DocumentProperties`和`SetPrinter`等函数修改`DEVMODE`结构，以改变纸张尺寸、纸张来源、打印质量和颜色模式。代码示例详细展示了整个配置过程，最后使设置生效。

摘要由CSDN通过智能技术生成

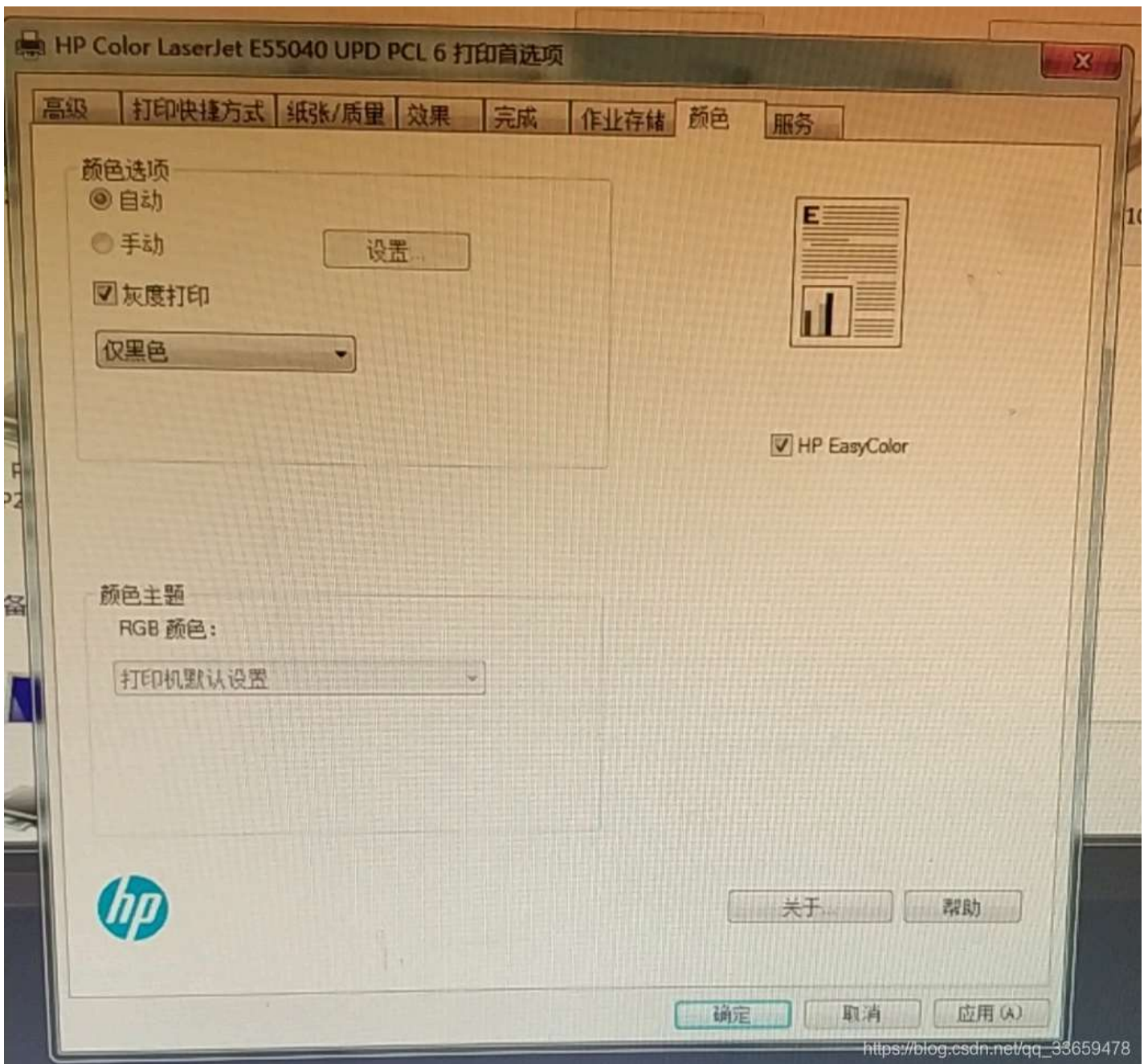
C++打印机首选项设置

前言

最近在开发打印机SDK，之前一直做黑白打印机，后面来了一个彩色打印机，要求彩色文档不但可以打印出彩色纸张，也能打印出黑白纸张；打印机默认是黑白文档打印黑白纸张，彩色文档打印彩色纸张，需要调用windows api来控制。

一、打印机首选项

首先查看打印机是否支持颜色设置（必须先安装打印机驱动，并且正常连接打印机），去电脑查看打印机首选项，【控制面板】->【设备和打印机】->【选中彩色打印机】->【右击选择打印机首选项】->【点击颜色窗口】，如下图所示：



PS: 每一台打印机的首选项界面不同, 只要找到颜色设置界面, 基本可以通过windows API来控制

二、打印流程

window打印都是右击文件点击打印, 使用打印机默认配置进行打印; 如果需要自定义打印, 打开打印机首选项对话框, 在里面可以设置纸张尺寸、纸张来源、纸张类型和单双面打印等等, 修改完之后, 再次选择文件打印, 就会按照修改后的打印机配置进行打印;

C++控制打印机打印的流程和windows 打印一样, 先通过Windows API来修改打印机首选项里面的配置

代码如下 (示例) :

```
1  #include <windows.h>
2  #include <string.h>
3
4  int SetPrintParams(void)
5  {
6      HANDLE hPrinter = NULL;
7      PRINTER_INFO_2* pi2 = NULL;
8      DEVMODE* pDevMode = NULL;
9      PRINTER_DEFAULTS pd;
10     DWORD dwNeeded = 0;
11
12 }
```

```

12     BOOL bFlag;
13     LONG lFlag;
14
15     // 获取默认打印机名称
16     CHAR szDevName[MAX_PATH] = "";
17     DWORD dwLength = MAX_PATH;
18     if (!GetDefaultPrinter(szDevName, &dwLength))
19     {
20         return 1;
21     }
22
23     ZeroMemory(&pd, sizeof(pd));
24     pd.DesiredAccess = PRINTER_ALL_ACCESS;
25     // 打开打印机
26     bFlag = OpenPrinter(szDevName, &hPrinter, &pd);
27     if (!bFlag || (hPrinter == NULL))
28     {
29         return 2;
30     }
31     // 获取pi2数据结构的尺寸
32     bFlag = GetPrinter(hPrinter, 2, 0, 0, &dwNeeded);
33     if ((!bFlag) && (GetLastError() != ERROR_INSUFFICIENT_BUFFER) || (dwNeeded == 0))
34     {
35         ClosePrinter(hPrinter);
36         hPrinter = NULL;
37         return 3;
38     }
39
40     pi2 = (PRINTER_INFO_2 *)GlobalAlloc(GPTR, dwNeeded);
41     if (pi2 == NULL)
42     {
43         ClosePrinter(hPrinter);
44         hPrinter = NULL;
45         return 4;
46     }
47
48     // 取得与指定打印机有关的信息PRINTER_INFO_2
49     bFlag = GetPrinter(hPrinter, 2, (LPBYTE)pi2, dwNeeded, &dwNeeded);
50     if (!bFlag)
51     {
52         GlobalFree(pi2);
53         ClosePrinter(hPrinter);
54         hPrinter = NULL;
55         return 5;
56     }
57
58     // DEVMODE数据结构中包含了有关设备初始化和打印机环境的信息
59     if (pi2->pDevMode == NULL)
60     {
61         // 获取pDevMode数据结构的尺寸
62         dwNeeded = DocumentProperties(NULL, hPrinter,
63             szDevName,
64             NULL, NULL, 0);
65         if (dwNeeded <= 0)
66         {
67             GlobalFree(pi2);
68             ClosePrinter(hPrinter);
69             hPrinter = NULL;
70             return 6;
71         }
72
73         pDevMode = (DEVMODE *)GlobalAlloc(GPTR, dwNeeded);
74         if (pDevMode == NULL)
75         {

```

```

76         GlobalFree(pi2);
77         ClosePrinter(hPrinter);
78         hPrinter = NULL;
79         return 7;
80     }
81
82     // 输出打印机设置信息到pDevMode
83     lFlag = DocumentProperties(NULL, hPrinter,
84         szDevName,
85         pDevMode, NULL,
86         DM_OUT_BUFFER);
87     if (lFlag != IDOK || pDevMode == NULL)
88     {
89         GlobalFree(pDevMode);
90         GlobalFree(pi2);
91         ClosePrinter(hPrinter);
92         hPrinter = NULL;
93         return 8;
94     }
95
96     pi2->pDevMode = pDevMode;
97 }
98
99 // 指定打印机的出纸匣, 每台打印机的纸匣号不一样, 我这边是, 0自动选择, 1是纸匣1, 2是纸匣2
100 pi2->pDevMode->dmFields = pi2->pDevMode->dmFields | DM_DEFAULTSOURCE;
101 int iBoxID = 0;
102 pi2->pDevMode->dmDefaultSource = iBoxID;
103
104 // 打印机使用的纸张尺寸(查找对应系统宏定义: DMPAPER_A4 9)
105 pi2->pDevMode->dmFields = pi2->pDevMode->dmFields | DM_PAPERSIZE;
106 int iPageSize = DMPAPER_A4;
107 pi2->pDevMode->dmPaperSize = iPageSize;
108
109 // 纸张类型, 根据不同打印机有不同配置, 我这边普通纸张是284
110 pi2->pDevMode->dmFields = pi2->pDevMode->dmFields | DM_MEDIATYPE;
111 int iMediaType = 284;
112 pi2->pDevMode->dmMediaType = iMediaType;
113
114 // 双面打印模式
115 pi2->pDevMode->dmFields = pi2->pDevMode->dmFields | DM_DUPLEX;
116 int iOffset = pi2->pDevMode->dmSize;
117 LPBYTE lpByteDevMode = (LPBYTE)pi2->pDevMode;
118 // 是否双面(1单面 / 2双面)
119 int iDuplex = 1;
120 if (iDuplex == 1)
121 {
122     pi2->pDevMode->dmDuplex = 1;
123     lpByteDevMode[iOffset + 84] = 1;
124     lpByteDevMode[iOffset + 86] = 0;
125 }
126 else if (iDuplex == 2)
127 {
128     pi2->pDevMode->dmDuplex = 2;
129     lpByteDevMode[iOffset + 84] = 2;
130     lpByteDevMode[iOffset + 86] = 1;
131 }
132
133 // 彩色打印机颜色设置
134 pi2->pDevMode->dmFields = pi2->pDevMode->dmFields | DM_COLOR;
135 // 1单色, 2彩色打印
136 int iColor = 1;
137 if (iColor == 1)
138 {
139     pi2->pDevMode->dmColor = DMCOLOR_MONOCHROME;
140

```

```

141     }
142     else if (iColor == 2)
143     {
144         pi2->pDevMode->dmColor = DMCOLOR_COLOR;
145     }
146
147     pi2->pSecurityDescriptor = NULL;
148
149     // 将修改后的pDevMode载入打印机驱动程序的新位置
150     lFlag = DocumentProperties(NULL, hPrinter,
151         szDevName,
152         pi2->pDevMode,
153         pi2->pDevMode,
154         DM_IN_BUFFER | DM_OUT_BUFFER);
155     if (lFlag != IDOK)
156     {
157         GlobalFree(pi2);
158         ClosePrinter(hPrinter);
159         if (pDevMode)
160         {
161             GlobalFree(pDevMode);
162         }
163         hPrinter = NULL;
164         return 10;
165     }
166
167     // 将pi2载入打印机
168     bFlag = SetPrinter(hPrinter, 2, (LPBYTE)pi2, 0);
169     if (!bFlag)
170     {
171         GlobalFree(pi2);
172         ClosePrinter(hPrinter);
173         if (pDevMode)
174         {
175             GlobalFree(pDevMode);
176         }
177         hPrinter = NULL;
178         return 11;
179     }
180
181     // 使打印机配置修改生效
182     SendMessageTimeout(HWND_BROADCAST, WM_DEVMODECHANGE,
183         0L,
184         (LPARAM)(LPCSTR)szDevName,
185         SMT0_NORMAL, 1000, NULL);
186
187     // 关闭打印机释放资源
188     if (hPrinter != NULL)
189     {
190         int iC = ClosePrinter(hPrinter);
191         hPrinter = NULL;
192
193         if (pi2 != NULL)
194         {
195             GlobalFree(pi2);
196             pi2 = NULL;
197         }
198
199         if (pDevMode != NULL)
200         {
201             GlobalFree(pDevMode);
202             pDevMode = NULL;
203         }
204     }

```

```
205  
206     return 0;  
}
```

总结

以上代码就是打印机首选项里面部分参数的配置，实际使用可以通过传参进行控制，我这里都是写死参数，其他参数根据自己需要进行添加，主要是修改pDevMode的属性，附一个pDevMode格式说明，如下：

```
1 //PDevMode = _devicemodeW;  
2 // _devicemodeW = record  
3 //     dmDeviceName: array[0..CCHDEVICENAME - 1] of WideChar; {打印设备名称,唯一值}  
4 //     dmSpecVersion: Word; {DevMode的版本号}  
5 //     dmDriverVersion: Word; {打印驱动程序版本号}  
6 //     dmSize: Word; {DevMode结构的字节大小}  
7 //     dmDriverExtra: Word; {dmDriverExtra成员结构的大小}  
8 //     dmFields: DWORD; {DevMode结构中哪些成员已被初始化}  
9 //     dmOrientation: SHORT; {纸张方向;1(纵向)2(横向)}  
10 //     dmPaperSize: SHORT; {纸张大小}  
11 //     dmPaperLength: SHORT; {自定义纸张的高度,PageSize为0或256时有效}  
12 //     dmPaperWidth: SHORT; {自定义纸张的宽度,PageSize为0或256时有效}  
13 //     dmScale: SHORT; {打印输出的缩放系数}  
14 //     dmCopies: SHORT; {打印份数}  
15 //     dmDefaultSource: SHORT; {送纸的默认纸盒}  
16 //     dmPrintQuality: SHORT; {打印机的分辨率;4-高品质 3-中品质 2-低品质 1-草稿}  
17 //     dmColor: SHORT; {打印输出为彩色或是单色;1-彩色 2-单色}  
18 //     dmDuplex: SHORT; {纸张是否为双面打印输出;1-单面 2-水平 3-垂直(需打印支持)}  
19 //     dmYResolution: SHORT; {Y方向分辨率;单位:点}  
20 //     dmTTOption: SHORT; {如何打印TrueType字体}  
21 //     dmCollate: SHORT; {多份时,是否逐份打印}  
22 //     dmFormName: array[0..CCHFORMNAME - 1] of WideChar; {纸张名称}  
23 //     dmLogPixels: Word; {}  
24 //     dmBitsPerPel: DWORD; {显示设备的颜色分辨率;单位:像素}  
25 //     dmPelsWidth: DWORD; {可见设备表面(屏幕或打印机)的宽度}  
26 //     dmPelsHeight: DWORD; {可见设备表面(屏幕或打印机)的高度}  
27 //     dmDisplayFlags: DWORD; {设备的显示模式}  
28 //     dmDisplayFrequency: DWORD; {显示设备在特定模式下的频率}  
29 //     dmICMMethod: DWORD; {}  
30 //     dmICMIntent: DWORD; {}  
31 //     dmMediaType: DWORD; {}  
32 //     dmDitherType: DWORD; {}  
33 //     dmICCManufacturer: DWORD; {}  
34 //     dmICCModel: DWORD; {}  
35 //     dmPanningWidth: DWORD; {}  
36 //     dmPanningHeight: DWORD; {}  
37
```

设置完打印机首选项配置之后，就可以调用打印机进行打印了，我用的是Shell打印，这里不讲打印实现

另外，每台打印机首选项里面的属性配置值不一样，参数设置值不能通用，可以在调用 [函数](#) DocumentProperties（输出打印机设置信息到pDevMode）的时候，通过pDevMode将默认的打印机设置信息记录下来，这个就是打印机首选里面参数；然后修改打印机首选项里面的配置，再通过DocumentProperties获取；反复如此，最终该打印机对应的属性值。