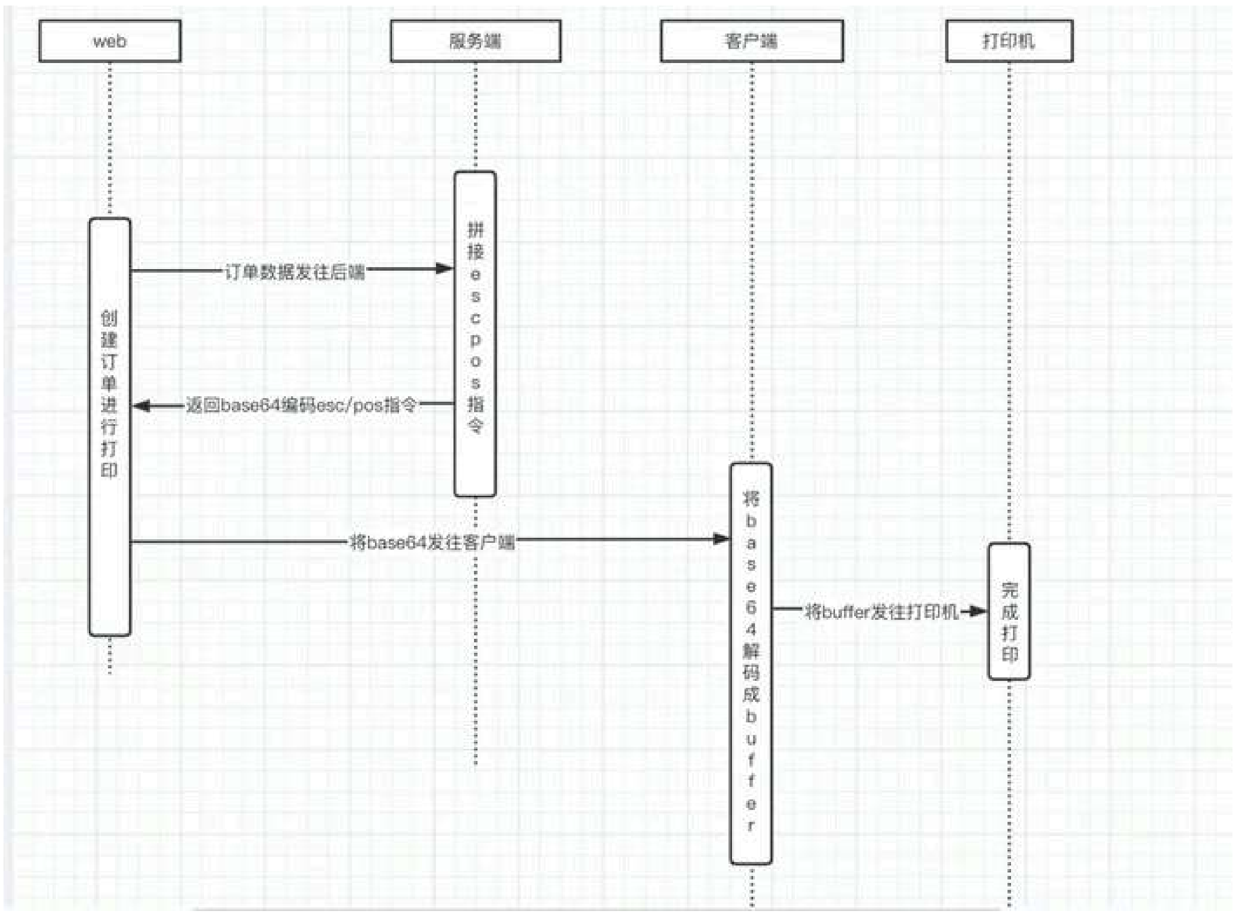
【转载】使用c++调用windows打印api进行打印的示例代码

原文： https://www.jb51.net/article/189553.htm

前言

在近期开发的收银台项目中，需要使用打印机进行小票打印，打印流程的时序图如下所示：



在客户的使用过程中，遇到一个问题，如果机器安装了打印机驱动，那么调用厂商提供的 sdk 进行打印的话，会导致出现小票只打印一半的情况，对此，需要绕过厂商 sdk 使用系统的打印才能够解决这一问题。

在 web 端打印中，需要调用浏览器打印 api 进行网页打印。这意味着，之前后端编写的esc/pos无法复用到，同时，前端还得花费精力来编写 html 以及css 来完成打印内容的排版，这无疑增加了复杂度以及工作量。正打算开始时，得到高人指点。

可以使用 windows api 进行打印

具体参见这篇文档

于是开始这方面的研究，功夫不负有心人，使用 windows api 完成了系统的打印，于是编写这篇文章记录踩过的坑。
首先看看如何进行打印：

```
BOOL RawDataToPrinter(LPSTR szPrinterName, LPBYTE lpData, DWORD dwCount)
{
  HANDLE    hPrinter;
  DOC_INFO_1 DocInfo;
  DWORD    dwJob;
  DWORD    dwBytesWritten;

  // Need a handle to the printer.
  if (!OpenPrinter(szPrinterName, &hPrinter, NULL)) {
      int y = GetLastError();
      cout << "openFail" << y << endl;
      return FALSE;
    }

    // Fill in the structure with info about this "document."

    DocInfo.pDocName = LPSTR("My Document\0");
    DocInfo.pOutputFile = NULL;
    DocInfo.pDatatype = NULL; // LPWSTR("RAW\0");
    // Inform the spooler the document is beginning.
    if ((dwJob = StartDocPrinter(hPrinter, 1, (LPBYTE)&DocInfo)) == 0)
    {
      int x = GetLastError();
      cout << "StartDocPrinter Fail" << x << endl;
      ClosePrinter(hPrinter);
      return FALSE;
    }
    // Start a page.
    if (!StartPagePrinter(hPrinter))
    {
      EndDocPrinter(hPrinter);
      ClosePrinter(hPrinter);
      return FALSE;
    }
    // Send the data to the printer.
    if (!WritePrinter(hPrinter, lpData, dwCount, &dwBytesWritten))
    {
      EndPagePrinter(hPrinter);
      EndDocPrinter(hPrinter);
      ClosePrinter(hPrinter);
      return FALSE;
    }
    // End the page.
    if (!EndPagePrinter(hPrinter))
    {
      EndDocPrinter(hPrinter);
      ClosePrinter(hPrinter);
      return FALSE;
    }
    // Inform the spooler that the document is ending.
    if (!EndDocPrinter(hPrinter))
    {
      ClosePrinter(hPrinter);
      return FALSE;
    }
    // Tidy up the printer handle.
    ClosePrinter(hPrinter);
    // Check to see if correct number of bytes were written.
    if (dwBytesWritten != dwCount)
      return FALSE;
    return TRUE;
  }
```

在文档中提到，打开打印机时"OpenPrinter"可以传入 null 以使用本地打印服务，因为不知道打印机名称，于是就传入了 null，结果在 StartDocPrinter 时一直提示失败，后来了解到使用 GetLastError 可以查看 error code，得到错误码后一对照，发现是 handle 是无效的，也就意味这 OpenPrinter 这一步骤没有打开需要的打印机。于是尝试使用 设备与打印机中的打印机名称，还真就连上了，成功调用打印服务。

但客户电脑上的打印机名称是不固定的，不能使用固定打印机名称，所以得拿到已经连接了的打印机列表，于是搜索到了 EnumPrinters 这一api，具体用法如下：

```cpp
void getPrinterList() {
  PRINTER_INFO_2* printerList;
  unsigned char size;
  unsigned long pcbNeeded;
  unsigned long pcReturned;

  EnumPrinters(PRINTER_ENUM_LOCAL, NULL, 2, NULL, 0, &pcbNeeded, &pcReturned);

  if ((printerList = (PRINTER_INFO_2*)malloc(pcbNeeded)) == 0) {
    return;
  }

  if (!EnumPrinters(PRINTER_ENUM_LOCAL, NULL, 2, (LPBYTE)printerList, pcbNeeded, &pcbNeeded, &pcReturned)) {
    free(printerList);
    return;
  }

  for (int i = 0; i < (int)pcReturned; i++) {

    string printName(printerList[i].pPrinterName);
    if (printerList[i].Attributes & PRINTER_ATTRIBUTE_NETWORK) {
      cout << "网络打印机" << printName << endl;
    }
    else {
      cout << "本地打印机" << printName << endl;
    }
  }

  cout << "number " << pcReturned << endl;

}
```

通过这一方式，的确获取到了系统中可用的打印机，可是拿到可用的打印机后还是有一个问题：“如何知道哪一个是小票打印机”？

为此又进行了搜索，又找到了一个 api GetDefaultPrinter，用法如下：

```cpp
string getDefaultPrinterName() {
  DWORD size = 0;
  GetDefaultPrinter(NULL, &size);

  if (size) {
    TCHAR* buffer = new TCHAR[size];
    GetDefaultPrinter(buffer, &size);
    string printerName(buffer);
    return printerName;
  }
  else {
    return "";
  }
}
```

通过此方法获取到系统默认打印机，客户只需要设置默认的打印机为小票打印机就完美解决问题了。

以下是完整代码：

```
#include <iostream>
#include <windows.h>
#include "node.h"
#include "base64.h"

using namespace std;
using v8::FunctionCallbackInfo;
using v8::Isolate;
using v8::Local;
using v8::NewStringType;
using v8::Object;
using v8::String;
using v8::Value;
using v8::Integer;
using v8::Int8Array;

BOOL RawDataToPrinter(LPSTR szPrinterName, LPBYTE lpData, DWORD dwCount);
string getDefaultPrinterName();

void localPrintRawData(const FunctionCallbackInfo<Value>& args) {
  Isolate* isolate = args.GetIsolate();
  Local<v8::Context> context = isolate->GetCurrentContext();
  v8::String::Utf8Value portString(isolate, args[0]);
  std::string base64Str(*portString);

  vector<BYTE> bytes = base64_decode(base64Str);
  char* buffer = new char[bytes.size()];
  copy(bytes.begin(), bytes.end(), buffer);
  string printerName = getDefaultPrinterName();
  if (printerName.size() > 0) {
    printerName += "\0";
    wstring ws(printerName.begin(), printerName.end());
    RawDataToPrinter(const_cast<char*>(printerName.c_str()), &bytes[0], bytes.size());
  }
  else {
    cout << "no printer" << endl;
  }
}

BOOL RawDataToPrinter(LPSTR szPrinterName, LPBYTE lpData, DWORD dwCount)
{
  HANDLE   hPrinter;
  DOC_INFO_1 DocInfo;
  DWORD   dwJob;
  DWORD   dwBytesWritten;

  // Need a handle to the printer.
  if (!OpenPrinter(szPrinterName, &hPrinter, NULL)) {
    int y = GetLastError();
```

```cpp
    cout << "openFial" << y << endl;
    return FALSE;
  }

  // Fill in the structure with info about this "document."

  DocInfo.pDocName = LPSTR("My Document\0");
  DocInfo.pOutputFile = NULL;
  DocInfo.pDatatype = NULL; // LPWSTR("RAW\0");
  // Inform the spooler the document is beginning.
  if ((dwJob = StartDocPrinter(hPrinter, 1, (LPBYTE)&DocInfo)) == 0)
  {
    int x = GetLastError();
    cout << "StartDocPrinter Fial" << x << endl;
    ClosePrinter(hPrinter);
    return FALSE;
  }
  // Start a page.
  if (!StartPagePrinter(hPrinter))
  {
    EndDocPrinter(hPrinter);
    ClosePrinter(hPrinter);
    return FALSE;
  }
  // Send the data to the printer.
  if (!WritePrinter(hPrinter, lpData, dwCount, &dwBytesWritten))
  {
    EndPagePrinter(hPrinter);
    EndDocPrinter(hPrinter);
    ClosePrinter(hPrinter);
    return FALSE;
  }
  // End the page.
  if (!EndPagePrinter(hPrinter))
  {
    EndDocPrinter(hPrinter);
    ClosePrinter(hPrinter);
    return FALSE;
  }
  // Inform the spooler that the document is ending.
  if (!EndDocPrinter(hPrinter))
  {
    ClosePrinter(hPrinter);
    return FALSE;
  }
  // Tidy up the printer handle.
  ClosePrinter(hPrinter);
  // Check to see if correct number of bytes were written.
  if (dwBytesWritten != dwCount)
    return FALSE;
  return TRUE;
}

void getPrinterList() {
  PRINTER_INFO_2* printerList;
  unsigned char size;
  unsigned long pcbNeeded;
  unsigned long pcReturned;

  EnumPrinters(PRINTER_ENUM_LOCAL, NULL, 2, NULL, 0, &pcbNeeded, &pcReturned);

  if ((printerList = (PRINTER_INFO_2*)malloc(pcbNeeded)) == 0) {
    return;
  }

  if (!EnumPrinters(PRINTER_ENUM_LOCAL, NULL, 2, (LPBYTE)printerList, pcbNeeded, &pcbNeeded, &pcReturned)) {
    free(printerList);
    return;
  }

  for (int i = 0; i < (int)pcReturned; i++) {

    string printName(printerList[i].pPrinterName);
    if (printerList[i].Attributes & PRINTER_ATTRIBUTE_NETWORK) {
      cout << "网络打印机" << printName << endl;
    }
    else {
      cout << "本地打印机" << printName << endl;
    }
  }

  cout << "number " << pcReturned << endl;

}

string getDefaultPrinterName() {
  DWORD size = 0;
  GetDefaultPrinter(NULL, &size);

  if (size) {
    TCHAR* buffer = new TCHAR[size];
    GetDefaultPrinter(buffer, &size);
    string printerName(buffer);
    return printerName;
  }
  else {
    return "";
  }
}

void Initialize(Local<Object> exports) {
  NODE_SET_METHOD(exports, "localPrintRawData", localPrintRawData);
}

NODE_MODULE(zq_device, Initialize)
```

参考:

https://support.microsoft.com/zh-cn/help/138594/howto-send-raw-data-to-a-printer-by-using-the-win32-api

https://docs.microsoft.com/en-us/windows/win32/printdocs/openprinter

https://stackoverflow.com/questions/6682286/understanding-a-c-sample-printers-handles-strings

https://social.msdn.microsoft.com/Forums/windowsdesktop/en-US/a27c6615-9452-44b1-90fc-9b91b15f0e50/openprinter-returing-errorinvalidprintername1801-when-called-with?forum=windowsgeneraldevelopmentissues

https://social.msdn.microsoft.com/Forums/vstudio/en-US/de7c55a1-ae63-49c9-a87a-fe3bf32822e4/how-to-use-the-enumprinters-function-to-be-able-to-classify-installed-printers-into-quot-network?forum=vclanguage

https://docs.microsoft.com/en-us/windows/win32/debug/system-error-codes--0-499-

https://docs.microsoft.com/zh-cn/windows/win32/debug/system-error-codes--1700-3999-?redirectedfrom=MSDN