Software Developer's Kit for Virtual Image Printer

1. FULL Version
After you purchase a developer’s license, you will receive a full version by email.

2. Source Code
You will get full source code including all files.
Entire code is written on C/C++.
For compilation you will need WINDDK 3790.1830 or more and Microsoft Visual Studio C++ 2005 Pro or more.
Inno Setup is required for installer compilation. (file Setup.iss)

3. Integrate into your own installer.
Even if you buy without source code, you will get installer script (written using Inno setup).
It allows you to integrate Virtual Image Printer into your own installer.

4. Silent installation
Launch setup.exe /silent to start installation in silent mode, without any notifications.

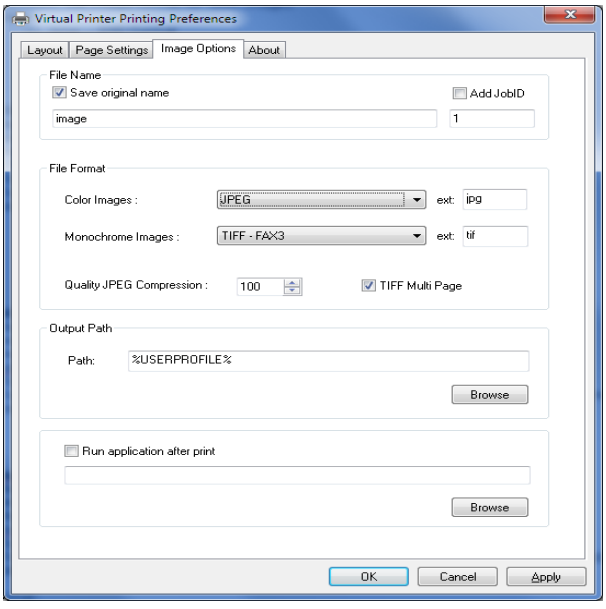
5. Custom Printer Name
Inform us the printer name you want and you'll get a package with it.

6. Development Interface for developers
The following settings in the registry hive control the behavior of Virtual Image Printer.
Before you decide whether to purchase a developer’s license of Virtual Image Printer, you may control Virtual Image Printer programmatically by yourself to evaluate the features of Virtual Image Printer.

All printer settings are stored in Windows registry:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter

Show Option:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
show_option
Type: STRING
Value:
1)true
2)false

Function: Hide or show options:



Output Path:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
path
Type: STRING
Function: Sets the default target folder for output print-ready files.

Color Format:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
color_format
Type: STRING
Function: Set image format for color images
Value:

1) bmp – BMP Format
2) jpg – JPEG Format
3) png – PNG Format
4) gif – GIF Format
5) zip – TIFF Format with zip compression
6) lzw – TIFF Format with lzw compression
7) t_jpeg – TIFF Format with jpeg compression
8) packbits – TIFF Format with packbits compression

B/W Format:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
bw_format
Type: STRING
Function: Set image format for b/w images
Value:

1) bmp – BMP Format
2) jpg – JPEG Format
3) png – PNG Format
4) gif – GIF Format
5) zip – TIFF Format with zip compression
6) lzw – TIFF Format with lzw compression
7) packbits – TIFF Format with packbits compression
8) fax3 – TIFF Format with fax3 compression
9) fax4 – TIFF Format with fax4 compression

ADD JobID:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
add_jobid
Type: STRING
Function: Value indicates that the printing job's ID will be added at the end of each print-ready file's name
Value:

1)true
2)false

Current JobID:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
current_jobid
Type: STRING
Function: Current JobID

Multi page fot TIFF Format:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
multipage
Type: STRING
Function: Value multipage indicates that each initial document will be converted into a single page of the TIFF file rather than separate TIFF file

Value:
1)true
2)false

Run application after printing :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
load_extapp
Type: STRING
Function: Opens the print-ready documents in the corresponding application when the printing is over.

Value:
1)true
2)false

Sample: "C:\Documents and Settings\All Users\test\C_aaaa_1.png" "C:\Documents and Settings\All Users\test\C_aaaa_2.png"

Choose the application after printing :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
extapp
Type: STRING
Function: Select the application after printing

Value: Full File Name.

Note: Value “load_extapp” must be “true”

Save original file name :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
originalname
Type: STRING
Function: Your print-ready file will have the same name as the initial document.

Value:
1)true
2)false

File Name :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
image_name
Type: STRING

Function: Specified the file name

Note: Value “originalname” must be “false”

JPEG Quality of image compression :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
q_jpg
Type: STRING
Function: Quality of image compression list box enables you to set the quality of the output file in percent (100 stands for the best possible quality

Value: 1 – 100

File extension for color images :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
color_ext
Type: STRING
Function: Set file extension for color images

File extension for B/W images :
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
bw_ext
Type: STRING
Function: Set file extension for b/w images

Send WM_COPYDATA Message to your application:
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
WM_FORM_NAME
Type: STRING
Class name or window name of your application.
Function: Call Windows API FindWindow(using "WM_FORM_NAME") to get HWND and fill out COPYDATASTRUCT Structure, then send WM_COPYDATA message to your application.

```
struct message_data{
        int job_id;
        wchar_t str[MAX_PATH * 4];
};
```

job_id = current jobid
wchar_t str[MAX_PATH * 4] = current print file name.

Function is not called if parameter (WM_FORM_NAME) is empty

This code sample demonstrates receiving a WM_COPYDATA from the printer:

C++

```
struct MY_STRUCT
{
   int Number;
   wchar_t Message[MAX_PATH*4];
};

BOOL OnCopyData(HWND hWnd, HWND hwndFrom, PCOPYDATASTRUCT pcds)
{
   MY_STRUCT myStruct;

   if (pcds->cbData == sizeof(myStruct))
   {
      memcpy_s(&myStruct, sizeof(myStruct), pcds->lpData, pcds->cbData);

      // Display the MY_STRUCT value in the window.
      SetDlgItemInt(hWnd, IDC_NUMBER_STATIC, myStruct.Number, TRUE);
      SetDlgItemText(hWnd, IDC_MESSAGE_STATIC, myStruct.Message);
   }

   return TRUE;
}
```

C#

```
      protected override void WndProc(ref Message m)
      {
         if (m.Msg == WM_COPYDATA)
         {
            // Get the COPYDATASTRUCT struct from lParam.
            COPYDATASTRUCT cds = (COPYDATASTRUCT)m.GetLParam(typeof(COPYDATASTRUCT));

            // If the size matches
            if (cds.cbData == Marshal.SizeOf(typeof(MyStruct)))
            {
               // Marshal the data from the unmanaged memory block to a
               // MyStruct managed struct.
               MyStruct myStruct = (MyStruct)Marshal.PtrToStructure(cds.lpData,
                  typeof(MyStruct));

               // Display the MyStruct data members.
               this.lbNumber.Text = myStruct.Number.ToString();
               this.lbMessage.Text = myStruct.Message;
            }
         }

         base.WndProc(ref m);
      }


      [StructLayout(LayoutKind.Sequential, CharSet = CharSet.Unicode)]
      internal struct MyStruct
      {
         public int Number;

         [MarshalAs(UnmanagedType.ByValTStr, SizeConst = MAH_PATH*4)]
         public string Message;
      }
```

Create log file
Registry Key:
HKEY_LOCAL_MACHINE\SOFTWARE\VirtualPrinter
Registry Key Value:
is_log_file
Type: STRING
false = Do not create log.
true = Create log file.

Registry Key Value:
log_file
Type: STRING
log file name with full path.