

原 小试ImageMagik——开发篇

2014年05月24日 00:11:13 阅读数：8339

=====

ImageMagick的使用和开发的文章：

[小试ImageMagik——使用篇](#)

[小试ImageMagik——开发篇](#)

=====

本文介绍使用ImageMagick开发程序的方法。ImageMagick安装之后就可以支持C/C++程序的开发，提供了3种接口。在这里首先介绍一下ImageMagick的3种接口。

MagickCore：

底层的C语言接口。较复杂，但是可以修改很多参数，只适合高端用户使用。

MagickWand：

推荐的C语言接口。相比于MagickCore接口，简单很多。适合普通用户使用。

Magick++：

提供面向对象的C++接口。

下面回顾一下ImageMagick安装后目录：

其中几个和开发有关的文件：

Lib文件夹：开发需要使用的静态库文件。包含4个库，前3个对应ImageMagick的3个接口：

CORE_RL_magick_.lib; CORE_RL_Magick++_.lib; CORE_RL_wand_.lib; X11.lib;

Include文件夹：开发需要使用的头文件。包含3个文件夹，对应ImageMagick的3个接口：

magick; Magick++; wand;

***.dll：**开发和使用过程中需要使用的动态链接库文件。

在开发中我们需要3种文件：头文件（*.h），静态库文件（*.lib），动态库文件（*.dll）。因此我们在VC中新建一个工程，然后将Lib文件夹，Include文件夹，以及dll拷贝到工程目录下，并且配置一下头文件和静态库的路径，就可以了。

下面分别给出ImageMagick的3种接口对应的例子。

MagickCore（底层的C语言接口。较复杂，但是可以修改很多参数，只适合高端用户使用）

功能：读取文件，创建缩略图，并保存成文件。

```

1.  /*
2.  * 雷霄骅
3.  * leixiaohua1020@126.com
4.  * http://blog.csdn.net/leixiaohua1020
5.  * 中国传媒大学/数字电视技术
6.  */
7.  #include <stdio.h>
8.  #include <stdlib.h>
9.  #include <string.h>
10. #include <time.h>
11. #include <magick/MagickCore.h>
12.
13. int main(int argc, char **argv)
14. {
15.     ExceptionInfo
16.     *exception;
17.
18.     Image
19.     *image,
20.     *images,
21.     *resize_image,
22.     *thumbnails;
23.
24.     ImageInfo
25.     *image_info;
26.
27.     if (argc != 3)
28.     {
29.         (void) fprintf(stdout, "Usage: %s image thumbnail\n", argv[0]);
30.         exit(0);
31.     }
32.     /*
33.      Initialize the image info structure and read an image.
34.     */
35.     MagickCoreGenesis(*argv, MagickTrue);
36.     exception=AcquireExceptionInfo();
37.     image_info=CloneImageInfo((ImageInfo *) NULL);
38.     (void) strcpy(image_info->filename, argv[1]);
39.     images=ReadImage(image_info, exception);
40.     if (exception->severity != UndefinedException)
41.         CatchException(exception);
42.     if (images == (Image *) NULL)
43.         exit(1);
44.     /*
45.      Convert the image to a thumbnail.
46.     */
47.     thumbnails=NewImageList();
48.     while ((image=RemoveFirstImageFromList(&images)) != (Image *) NULL)
49.     {
50.         resize_image=ResizeImage(image, 106, 80, LanczosFilter, 1.0, exception);
51.         if (resize_image == (Image *) NULL)
52.             MagickError(exception->severity, exception->reason, exception->description);
53.         (void) AppendImageToList(&thumbnails, resize_image);
54.         DestroyImage(image);
55.     }
56.     /*
57.      Write the image thumbnail.
58.     */
59.     (void) strcpy(thumbnails->filename, argv[2]);
60.     WriteImage(image_info, thumbnails);
61.     /*
62.      Destroy the image thumbnail and exit.
63.     */
64.     thumbnails=DestroyImageList(thumbnails);
65.     image_info=DestroyImageInfo(image_info);
66.     exception=DestroyExceptionInfo(exception);
67.     MagickCoreTerminus();
68.     return(0);
69. }

```

MagickWand（推荐的C语言接口。相比于MagickCore接口，简单很多。适合普通用户使用）

功能：读取文件，创建缩略图，并保存成文件。

```

1.  /*
2.   * 雷霄骅
3.   * leixiaohua1020@126.com
4.   * http://blog.csdn.net/leixiaohua1020
5.   * 中国传媒大学/数字电视技术
6.   */
7.  #include <stdio.h>
8.  #include <stdlib.h>
9.  #include <wand/MagickWand.h>
10.
11. int main(int argc, char **argv)
12. {
13.
14.     MagickBooleanType status;
15.
16.     MagickWand *magick_wand;
17.
18.     if (argc != 3)
19.     {
20.         (void) fprintf(stdout, "Usage: %s image thumbnail\n", argv[0]);
21.         exit(0);
22.     }
23.     /*
24.      Read an image.
25.     */
26.     MagickWandGenesis();
27.     magick_wand = NewMagickWand();
28.     status = MagickReadImage(magick_wand, argv[1]);
29.     /*
30.      Turn the images into a thumbnail sequence.
31.     */
32.     MagickResetIterator(magick_wand);
33.     while (MagickNextImage(magick_wand) != MagickFalse)
34.     {
35.         MagickResizeImage(magick_wand, 106, 80, LanczosFilter, 1.0);
36.     }
37.     /*
38.      Write the image then destroy it.
39.     */
40.     status = MagickWriteImages(magick_wand, argv[2], MagickTrue);
41.
42.     magick_wand = DestroyMagickWand(magick_wand);
43.     MagickWandTerminus();
44.     return(0);
45. }

```

Magick++（提供面向对象的C++接口）

```



1.  /*
2.   * 雷霄骅
3.   * leixiaohua1020@126.com
4.   * http://blog.csdn.net/leixiaohua1020
5.   * 中国传媒大学/数字电视技术
6.   */
7.  //创建Image对象,
8.  // create a blank image canvas with 640x480 size and 'white' color as background:
9.  Image blank_image( Geometry(640, 480), Color(MaxRGB, MaxRGB, MaxRGB, 0));
10. // or also, by using the automatic C++ type conversions for the arguments:
11. Image blank_image("640x480", "white");
12.
13. // create an image from URL
14. Image url_image("http://www.serverName.com/image.gif");
15. Image local_file_image("my_image.gif"); // here the URL points to the local filesystem
16. //获取/设置属性
17. // Canvas geometry
18. // returns an unsigned int representing the my_image width
19. unsigned int Image::columns();
20. // returns an unsigned int representing the my_image height
21. unsigned int Image::rows();
22. // sets the my_image format; the format string can be "GIF", etc
23. void Image::magick("png");
24. // returns a string value representing the image format (e.g. "GIF", "JPEG", etc)
25. string Image::magick();
26. //读取/保存图像文件
27. // Reading the contents of a disk file into an image object can be performed
28. Image my_image(); // create an *empty* image using the default Image constructor
29. // read a GIF image file from disk; the image format is automatically set to GIF
30. my_image.read("aGIFImageFile.gif");
31. // Writing an Image object to a disk file. set the "format" attribute of my_image to PNG
32. my_image.magick("png");
33. // write to disk an image file
34. my_image.write("file_name_explicit_extension.gif");

```

MagickWand一般情况下属于使用比较普遍的，下面记录两个MagickWand开发的例子。

更多的例子可以参考：<http://members.shaw.ca/el.supremo/MagickWand/>

功能：将图像的宽高变成源图像的50%

```
[cpp]    
1.  /*  
2.  * 雷霄骅  
3.  * leixiaohua1020@126.com  
4.  * http://blog.csdn.net/leixiaohua1020  
5.  * 中国传媒大学/数字电视技术  
6.  */  
7.  #include <windows.h>  
8.  #include <wand/magick_wand.h>  
9.  
10. void test_wand(void)  
11. {  
12.     MagickWand *m_wand = NULL;  
13.  
14.     int width,height;  
15.  
16.     MagickWandGenesis();  
17.  
18.     m_wand = NewMagickWand();  
19.     // Read the image - all you need to do is change "logo:" to some other  
20.     // filename to have this resize and, if necessary, convert a different file  
21.     MagickReadImage(m_wand,"logo:");  
22.  
23.     // Get the image's width and height  
24.     width = MagickGetImageWidth(m_wand);  
25.     height = MagickGetImageHeight(m_wand);  
26.  
27.     // Cut them in half but make sure they don't underflow  
28.     if((width /= 2) < 1)width = 1;  
29.     if((height /= 2) < 1)height = 1;  
30.  
31.     // Resize the image using the Lanczos filter  
32.     // The blur factor is a "double", where > 1 is blurry, < 1 is sharp  
33.     // I haven't figured out how you would change the blur parameter of MagickResizeImage  
34.     // on the command line so I have set it to its default of one.  
35.     MagickResizeImage(m_wand,width,height,LanczosFilter,1);  
36.  
37.     // Set the compression quality to 95 (high quality = low compression)  
38.     MagickSetImageCompressionQuality(m_wand,95);  
39.  
40.     /* Write the new image */  
41.     MagickWriteImage(m_wand,"logo_resize.jpg");  
42.  
43.     /* Clean up */  
44.     if(m_wand)m_wand = DestroyMagickWand(m_wand);  
45.  
46.     MagickWandTerminus();  
47. }
```

功能：在图像的周围加上边框

```

1.  /*
2.   * 雷霄骅
3.   * leixiaohua1020@126.com
4.   * http://blog.csdn.net/leixiaohua1020
5.   * 中国传媒大学/数字电视技术
6.   */
7. #include <windows.h>
8. #include <wand/magick_wand.h>
9.
10. void test_wand(void)
11. {
12.     MagickWand *m_wand = NULL;
13.     PixelWand *p_wand;
14.     int w,h;
15.
16.     MagickWandGenesis();
17.
18.     /* Create a wand */
19.     m_wand = NewMagickWand();
20.     p_wand = NewPixelWand();
21.
22.     // Change this to whatever colour you like - e.g. "none"
23.     PixelSetColor(p_wand, "blue");
24.     /* Read the input image */
25.     MagickReadImage(m_wand,"logo:");
26.     w = MagickGetImageWidth(m_wand);
27.     h = MagickGetImageHeight(m_wand);
28.     MagickSetImageBackgroundColor(m_wand,p_wand);
29.     // This centres the original image on the new canvas.
30.     // Note that the extent's offset is relative to the
31.     // top left corner of the *original* image, so adding an extent
32.     // around it means that the offset will be negative
33.     MagickExtentImage(m_wand,1024,768,-(1024-w)/2,-(768-h)/2);
34.     MagickWriteImage(m_wand,"logo_extent.jpg");
35.
36.     /* Tidy up */
37.     m_wand = DestroyMagickWand(m_wand);
38.     p_wand = DestroyPixelWand(p_wand);
39.     MagickWandTerminus();
40. }

```

补充:详细的教程可以从ImageMagick的官网(<http://www.imagemagick.org/>)左侧的目录中查看。在Program Interfaces里面有几种接口的详细开发说明。

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/26754089>

文章标签： [ImageMagick](#) [开发](#) [c](#) [VC](#) [图像处理](#)

个人分类： [ImageMagick](#)

所属专栏： [开源多媒体项目源代码分析](#)

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com