

原创

老狼主

于 2019-12-23 23:40:13 发布

133

收藏

版权

分类专栏：

VC++ (日积月累篇)

VC++ (日积月累篇)

专栏收录该内容

9 订阅

203 篇文章

订阅专栏

```
1  #include <windows.h>
2  #include <gdiplus.h>
3  #include <stdio.h>
4  using namespace Gdiplus;
5  #pragma comment(lib, "gdiplus.lib")
6
7  // 获取JPEG图像编码器CLSID
8  int GetEncoderClsid(const WCHAR* format, CLSID* pClsid)
9  {
10     UINT num = 0;
11     UINT size = 0;
12     ImageCodecInfo* pImageCodecInfo = NULL;
13     //获取含有可用图像编码器的数目
14     GetImageEncodersSize(&num, &size);
15     if (size == 0)
16     {
17         return -1;
18     }
19
20     pImageCodecInfo = (ImageCodecInfo*)(malloc(size));
21     if (pImageCodecInfo == NULL)
22     {
23         return -1;
24     }
25
26     // 获取含有可用图像编码器信息的ImageCodecInfo对象数组，根据图像类型名称做对比，取出图像编码器CLSID
27     GetImageEncoders(num, size, pImageCodecInfo);
28     for (UINT j = 0; j < num; ++j)
29     {
30         if (wcscmp(pImageCodecInfo[j].MimeType, format) == 0)
31         {
32             *pClsid = pImageCodecInfo[j].Clsid;
33             free(pImageCodecInfo);
34             return j;
35         }
36     }
37
38     free(pImageCodecInfo);
39     return -1;
40 }
41 // 从内存加载图片，失败返回NULL
42 Bitmap* LoadBitmapFromMemory(const void* memory, DWORD size)
43 {
44     Bitmap* bmp = NULL;
45     IStream* stream = NULL;
46     // 从内存创建流对象
47     if (CreateStreamOnHGlobal(NULL, TRUE, &stream) == S_OK)
48     {
49         ULARGE_INTEGER uli;
50         uli.QuadPart = size;
```

- 分类专栏
- 工作随笔篇

172篇
- 算法

7篇
- VCPKG

1篇
- TightVNC

3篇
- CMake

10篇
- QT

4篇
- OSG

1篇
- OpenGL

7篇
- GLFW

7篇
- PE文件解析

12篇
- Docker
- Node.js

2篇
- Web前端开发

2篇
- HTML 教程

12篇
- CSS 教程

1篇
- JavaScript 教程

26篇
- JavaScript参考手册

23篇
- HTML 参考手册- (HTML5 标...
- Linux

5篇
- Nginx

15篇
- OpenCV

4篇
- Delphi 10

4篇
- FFmpeg

1篇
- C++ 语言教程

49篇
- EasyPR车牌识别

2篇
- C 语言教程

23篇
- VC++ (日积月累篇)

203篇
- VC++ 设计模式

2篇
- VC++ (基础篇)

61篇
- VC++ (GIS管理篇)

15篇
- VC++ (视频监控篇)

```

51 // 设置流大小
52 stream->SetSize(uli);
53 // 将流数据写入内存
54 if (stream->Write(memory, size, NULL) == S_OK)
55     // 从内存创建Bitmap图片
56     b m p = new Bitmap(stream);
57 stream->Release();
58 }
59
60 return bmp;
61 }
62 // 从文件加载图片, 不独占文件, 失败返回NULL
63 Bitmap* LoadBitmapFromFile(const TCHAR* file_name)
64 {
65     Bitmap* bmp = NULL;
66     // 打开图片文件
67     HANDLE file_handle = CreateFile(file_name, GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL);
68     if (file_handle != INVALID_HANDLE_VALUE)
69     {
70         DWORD temp = 0;
71         // 获取文件总大小
72         DWORD file_size = GetFileSize(file_handle, &temp);
73         if (file_size && !temp) // 不处理大于4G的文件
74         {
75             // 申请存放图片数据的内存空间
76             unsigned char* buffer = new unsigned char[file_size];
77             // 将图片文件读到内存后, 再从内存创建Bitmap
78             if (ReadFile(file_handle, buffer, file_size, &temp, NULL))
79                 b m p = LoadBitmapFromMemory(buffer, temp);
80             delete[] buffer;
81         }
82
83         // 关闭文件句柄
84         CloseHandle(file_handle);
85     }
86     else
87     {
88         DWORD dwErrorCode = GetLastError();
89         std::cout << "CreateFile fail. ErrorCode=" << dwErrorCode << endl;
90     }
91
92     return bmp;
93 }
94
95 // 更新文件版权信息
96 BOOL UpdataJpegExifCopyRightInfo(const wchar_t* filename, const char* exifCopyrightInfo)
97 {
98     BOOL bResult = FALSE;
99     // GDI+接口初始化
100     GdiplusStartupInput gdiplusStartupInput;
101     ULONG_PTR gdiplusToken;
102     GdiplusStartup(&gdiplusToken, &gdiplusStartupInput, NULL);
103
104     // 从文件名获取图片信息,
105     Bitmap* image = LoadBitmapFromFile(filename); // Bitmap::FromFile(filename); // Bitmap::FromFile这种方式文件是独占
106     if (image == NULL){
107         // 清理GDI+资源
108         GdiplusShutdown(gdiplusToken);
109         return bResult;
110     }
111

```

	VC++ (中级篇)	
	VC++ (高级篇)	
	VC++ (初级篇)	5篇
	VC++ (图像处理篇)	31篇
	VC++ (控件篇)	58篇
	VC++ (UI设计篇)	1篇
	VC++ (通讯篇)	1篇

```
112 image->GetPropertyItemSize(PropertyTagCopyright);
113
114 // 给EXIF属性申请内存空间
115 PropertyItem* propertyItem = (PropertyItem*)malloc(sizeof(PropertyItem));
116 memset(propertyItem, 0, sizeof(PropertyItem));
117 // 设置EXIF版本属性值
118 propertyItem->id = PropertyTagCopyright; // 版权信息对应的Tag值
119 propertyItem->type = PropertyTagTypeASCII; // 版权信息值类型
120 propertyItem->value = (void*)exifCopyrightInfo;
121 // 设置属性字符串长度
122 propertyItem->length = strlen(exifCopyrightInfo) + 1;
123
124 // 获取JPEG图像编码器CLSID
125 CLSID clsid;
126 GetEncoderClsid(L"image/jpeg", &clsid);
127
128 // 设置属性值
129 image->SetPropertyItem(propertyItem);
130 // 保存图片
131 Status status = image->Save(filename, &clsid, NULL);
132 if (status == Ok)
133     bResult = TRUE;
134
135 // 删除申请的内存空间
136 free(propertyItem);
137 delete image;
138
139 // 清理GDI+资源
140 GdiplusShutdown(gdiplusToken);
141
142 return bResult;
143 }
```

程序中调用的是GDI+接口来实现修改JPG图片的EXIF属性信息，切记打开文件不要用Bitmap::FromFile方式，此方式文件将会被占用，导致修改属性失败。

参与评论 您还未登录，请先 [登录](#) 后发表或查看评论

TightVNC H264编解码（一）

 310

经过尽二个月的不断摸索，TightVNC终于支持H264编解码了，前期真正编写H264编解码器只花了一周左右时间，但是测试发现效果并不是太理想（帧率太低，延迟大），这大概是所有远程控制软件要处理的通病吧。作为一位经验丰富的程序员，造成这种问题的原因，有以下几点：1、抓屏？2、H264编码？...

QTHttpServer

 270

资源从网上收集，包含httpserver、sslserver、http-parser三个模块，解决了编译过程中由于路径引用错误问题。下载链接：<https://download.csdn.net/download/u012156872/20664799>，下载编译即可应用到自己项目中，简单快速，不用造车。...

安防 海康摄像机视频存储服务器

 820

软件下载地址（附操作手册）：<https://download.csdn.net/download/u012156872/15501044>概述软件采用海康SDK接口设计，支持嵌入式和IPC网络摄像机。

QT 界面设计篇（水波纹进度条QProgressBarWater）

 666

效果图：头文件声明（QProgressBarWater.h）：

```
#ifndef QPROGRESSBARWATER_H#define QPROGRESSBARWATER_H#include <QWidget>#include <QPainter>#include <QPaintEvent>class QProgressBarWater : public QWidget{ Q_OBJECTpublic: explic.
```

EasyPR编译指南

 2323

1、下载源码<https://github.com/liuruoze/EasyPRDownloadZIP>或者git（<https://github.com/liuruoze/EasyPR.git>）将代码解压到硬盘目录下。例如F:\车牌识别\EasyPR（车牌识别）\EasyPR2、双击解决方案文件EasyPR.sln，用VSIDE（这里使用VS2019）打开，提示点击确定。3、选择libeasypr项目，编译出现...

GmSSL 在Windows上的使用（编译和使用）

 177



老狼主


企业

331	4729	5094	23万+	
原创	周排名	总排名	访问	等级

4795	3292	131	135	542
积分	粉丝	获赞	评论	收藏











私信

关注

搜博主文章 

热门文章

C 简介 9886

JavaScript RegExp 对象 9212

JavaScript 全局 7866

粤嵌GEC6818开发板-入门感慨篇 7102

Nginx access日志配置 3409

最新评论

MFC之 MapWinGIS叠加天地图瓦片地图
老狼主: 图层的顺序是可以调整的。

MFC之 MapWinGIS叠加天地图瓦片地图
老狼主: 图层的顺序是可以调整的。

MFC之 MapWinGIS叠加天地图瓦片地图
老狼主: 图层的顺序是可以调整的。

MFC之 MapWinGIS叠加天地图瓦片地图
我去额都无法: 您好, 能不能扩展加载WMT
S层. 现在瓦片地图只能放在最底层, 不能...

TightVNC H264编解码 (一)

weixin_39427165: 可以把H254直接改成H2
65吗, 进一步提高压缩率, 不少处理器也...

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

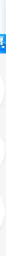
GmSSL 在Windows上的使用 (编译和使用)

VCPKG 包下载失败解决思路

TightVNC H264编解码 (三) 之去图像抖动

2022年 11篇 2021年 73篇

<div><div></div><div>VCPKG 包下载失败解决思路</div><div>1199</div></div> <div>vcpkg经常会遇到资源无法访问，可能是域名解析出了问题，我们只需要将域名解析后的ip地址添加到hosts文件列表中可解决此问题，如：185.199.108.133 raw.githubusercontent.com在此之前可通过终端ping指令验证一下。linux系统中， hosts文件路径： /etc/hostswindows系统中， hosts文件路径： C:\windo...</div>	<div><div></div><div>TightVNC H264编解码（三）之去图像抖动</div><div>369</div></div> <div>本以为服务器端改造完硬编码，客户端升级完解码功能之后，一切也就该收尾了，可没成想如鱼哽喉卡住了，客户端播放视频画面，当服务器端窗口有快速移动的时候，客户端显示窗口边缘会产生抖动（不够顺滑流畅），分析其原因：1、服务器端视频编码问题2、客户端视频解码问题3、网络数据丢包问题4...</div>
<div><div></div><div>macOS获取系统信息</div><div>205</div></div> <div>代码在XCode下编译通过类定义：<pre>#ifndef SYSTEMINFO_H#define SYSTEMINFO_H#include <string>#include <vector>using namespace std;namespace s hell { class StrUtil { public: static inline std::string& ltrim(std::string &str)</pre></div>	<div><div></div><div>C++之xlsx文件操作之写操作</div><div>444</div></div> <div>采用xlsxwriter开源库的实现，声明如下：<pre>#include ".xlsxwriter/include/xmlwriter.h"#include ".xlsxwriter/include/workbook.h"#include ".xlsxwriter/include/wor ksheet.h"#ifdef _DEBUG#pragma comment(lib, ".xlsxwriter/lib/debug/xlsxwriter.lib")#else#pragma comment(</pre></div>
<div><div></div><div>C++之xlsx文件操作之读操作</div><div>809</div></div> <div>采用xlint开源库的实现，声明如下：<pre>#define XLNT_STATIC 1#include ".xlint/xlint.hpp"#ifdef _DEBUG//pragma comment(lib, ".xlint/lib/xlntd.lib")#pragma co mment(lib, ".xlint/xlntd.lib")#else#pragma comment(lib, ".xlint/lib/xlint.lib")#endif // _DEBUGnamespace xlint{</pre></div>	<div><div></div><div>Delphi中TEdgeBrowser控件初探</div><div>423</div></div> <div>1、组件库安装Delphi IDE中的TEdgeBrowser虽然可以直接拖拽到界面 TForm上来，但是却是无法直接使用的，究其原因是因为缺少相关的库文件，这里介 绍最简单直接有效的方法：1、下载 nuget。nuget是一个开源项目的客户端下载工具，想要了解更多，请问度娘。2、打开终端，在命令行执行：.\nuget i...</div>
<div><div></div><div>C++连接Wlan/Wifi网络设备</div><div>560</div></div> <div>本单元只介绍如何链接wifi设备，对信号强度不做涉及，另外链接指定wifi之前应该先判断wifi是否已经链接，如果已经链接就没必要再链接了，代码中采用的是没做链接判断，先直接断开然后重新进行链接，代码如下：<pre>#include <wlanapi.h>#pragma comment(lib, "wlanapi.lib")BOOL connectWlan(CString Nam...</pre></div>	<div><div></div><div>已知文件被某进程占用，强行解除占用，并删除文件</div><div>230</div></div> <div><pre>BOOL GetVolumeNameByHandle(HANDLE hFile, LPTSTR szVolumeName, UINT cchMax){ BOOL bResult = FALSE; TCHAR szBuf[500] = { 0 }; TCHAR * pl ter = szBuf; int i=0; BY_HANDLE_FILE_INFORMATION stFileInfo = { 0 }; do { if(FALSE == GetFileInformationByHa.</pre></div>
<div><div></div><div>通过GDI+修改jpg文件EXIF属性</div><div>64</div></div> <div>代码直接从旧项目工程里面抠出来的，抛砖引玉。主函数名称<code>BOOL UpdateJpegExifCopyRightInfo(const wchar_t* filename, const char* exifCopyrightInfo)# include <windows.h>#include <gdiplus.h>#include <stdio.h>using namespace Gdiplus;#pragma comment(lib, "gdiplus.</code></div>	<div><div></div><div>TightVNC H264编解码（二）之硬编码库的编译</div><div>358</div></div> <div><pre>AVCodec * codec = avcodec_find_encoder_by_name("nvcnc_h264");如果是默认的ffmpeg库，返回结果是NULL，看来是不带有硬编码功能的，重新编译分 支ffnvcodec，不到半个小时编译完成，返回值终于ok了(^)，继续编写代码：AVCodecContext *avctx = avcodec_alloc_context3(codec);int ret = avco...</pre></div>
<div><div></div><div>CMake应用：基础篇</div><div>176</div></div> <div>CMake是一个开源、跨平台的编译、测试和打包工具，它使用比较简单的语言描述编译、安装的过程，输出Makefile或者project文件，再去执行构建。在使用IDE开发软件的过程中，代码的编译和构建一般是使用IDE自带的编译工具和环境进行编译，开发者参与的并不算多。如果想要控制构建的细节，则需要...</div>	<div><div></div><div>CMake应用：核心语法篇</div><div>94</div></div> <div>本文是深入CMakeLists.txt之前的前导文章，介绍CMake语言的核心概念，以及常用的CMake脚本命令，以期对CMake的语法能有比较好的认知和实践基础 。在前一篇文章中介绍了CMake的核心概念，使用的一般流程，并通过一个实例讲解了CMake命令行工具之一的cmake命令的使用方法。该系列文章请...</div>
<div><div></div><div>CMake应用：CMakeLists.txt完全指南</div><div>274</div></div> <div>CMake通过CMakeLists.txt配置项目的构建系统，配合使用cmake命令行工具生成构建系统并执行编译、测试，相比于手动编写构建系统（如Makefile）要高效许多。对于C/C++项目开发，非常值得学习掌握。在前两篇文章中已经介绍CMake的相关核心概念，使用的一般流程，以及CMake核心的语法和常用脚...</div>	<div><div></div><div>CMake应用：模块化及库依赖</div><div>359</div></div> <div>当项目比较大的时候，往往需要将代码划分为几个模块，可能还会分离出部分通用模块，在多个项目之间同时使用；当然，也可能是依赖开源的第三方库，在项目中包含第三方源代码或者编译好的库文件。本文将介绍CMake中如何模块化地执行编译，以及指定目标对相应库文件的依赖。在上一篇文章中，...</div>
“相关推荐”对你有帮助么？	
<div><div></div><div>非常没帮助</div><div></div><div>没帮助</div><div></div><div>一般</div><div></div><div>有帮助</div><div></div><div>非常有帮助</div></div>	
关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00	



2020年 204篇

2019年 142篇



老狼主

关注



1



0



0



专栏目录