

鸡啄米

聚焦互联网、数码、软件开发和编程入门的IT休闲吧

[首页](#) [IT互联网](#) [数码生活](#) [软件开发](#) [职场人生](#) [娱乐休闲](#) [编程课堂](#) [安卓开发](#) [留言簿](#)

[首页](#) » [软件开发](#) » MFC六大核心机制之一：MFC程序的初始化

MFC六大核心机制之一：MFC程序的初始化

分类标签: [MFC](#) [VC++](#)

很多做软件开发的人都有一种对事情刨根问底的精神，例如我们一直在用的MFC，很方便，不用学太多原理性的知识就可以做出各种窗口程序，但喜欢钻研的朋友肯定想知道，到底微软帮我们做了些什么，让我们在它的框架下可以简单的写程序。本文开始就跟大家分享一位同行前辈写的MFC核心机制分析（稍作整理），语言朴实易懂，在读完此深入浅析的剖析系列后，相信留给大家的是对MFC运行机制的深入理解。

MFC六大核心机制概述

我们选择了C++，主要是因为它够艺术、够自由，使用它我们可以实现各种想法，而MFC将多种可灵活使用的功能封装起来，我们岂能忍受这种“黑盒”操作？于是研究分析MFC的核心机制成为必然。

首先，列出要讲的MFC六大核心机制：

- 1、MFC程序的初始化。
- 2、运行时类型识别（RTTI）。
- 3、动态创建。
- 4、永久保存。
- 5、消息映射。
- 6、消息传递。

本文讲第一部分，MFC程序的初始化过程。

简单的MFC窗口程序

设计一个简单完整MFC程序，产生一个窗口。当然这不能让AppWizard自动为我们生成。我们可以在Win32 Application工程下面那样写：

c++代码

```
1. #include <afxwin.h>
2. class MyApp : public CWinApp
3. {
4. public:
5.     BOOL InitInstance() //②程序入点
6.     {
7.         CFrameWnd *Frame=new CFrameWnd(); //构造框架
8.         m_pMainWnd=Frame; //将m_pMainWnd设定为Frame;
9.         Frame->Create(NULL, "最简单的窗口"); //建立框架
10.        Frame->ShowWindow(SW_SHOW); //显示框架
11.        return true; //返回
12.    }
13. };
14. MyApp theApp; //①建立应用程序。
```

设定链接MFC库，运行，即可看见一个窗口。

从上面，大家可以看到建立一个MFC窗口很容易，只用两步：一是从CWinApp派生一个应用程

nat123映射全端口，全速外网访

nat123端口映射，内网映射外网，全端

公网加速不限速，73元/年起

nat123内网端口映射软件

订阅鸡啄米

[RSS](#) [+ 订阅到](#) [分享到](#)



一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

[百度分享](#)

站内搜索

请输入搜索内容..



分类标签

[编程入门 \(135\)](#)

[C++ \(96\)](#)

[VC++ \(77\)](#)

[MFC \(67\)](#)

[VS2010 \(63\)](#)

[程序员 \(55\)](#)

[Android \(51\)](#)

[Java \(51\)](#)

[苹果 \(49\)](#)

[智能手机 \(47\)](#)

[腾讯 \(41\)](#)

[百度 \(37\)](#)

[阿里巴巴 \(33\)](#)

[谷歌 \(32\)](#)

[平板电脑 \(31\)](#)

[TCP/IP \(29\)](#)

[iPhone \(26\)](#)

[PHP \(26\)](#)

[Javascript \(25\)](#)

[奇虎360 \(24\)](#)

[Mysql \(24\)](#)

[Windows \(22\)](#)

[软件架构 \(20\)](#)

[小米 \(20\)](#)

[设计模式 \(19\)](#)

[iPad \(18\)](#)

[Web \(18\)](#)

[职场攻略 \(18\)](#)

[三星 \(16\)](#)

[创业 \(16\)](#)

[微软 \(13\)](#)

[iOS \(13\)](#)

[微信 \(13\)](#)

[HTML \(13\)](#)

[应用程序 \(12\)](#)

序类（这里是MyApp），然后建立应用程序对象（theApp），就可以产生一个自己需要的窗口（即需要什么就在InitInstance()里创建就行了）。

整个程序，就改写一个InitInstance()函数，创建那么一个对象（theApp），就是一个完整的窗口程序。这就是“黑盒”操作的魔力！

在我们正想为微软鼓掌的时候，我们突然觉得心里空荡荡的，我们想知道微软帮我们做了什么事情，而我们想编自己的程序时又需要做什么事情，哪怕在上面几行的程序里面，我们还有不清楚的地方，比如，干嘛有一个m_pMainWnd指针变量，它从哪里来，又要到哪里去呢？想一想在DOS下编程是多么美妙的一件事呵，我们需要什么变量，就声明什么变量，需要什么样的函数，就编写什么样的函数，或者引用函数库……但是现在我们怎么办？

我们可以逆向思维一下，MFC要达到这种效果，它是怎么做呢？首先我们要弄明白，VC++不是一种语言，它就象我们学c语言的时候的一个类似记事本的编辑器（请原谅我的不贴切的比喻），所以，在VC里面我们用的是C++语言编程，C++才是根本（初学者总是以为VC是一门什么新的什么语言，一门比C++先进很多的复杂语言，汗）。说了那么多，我想用一句简单的话概括“MFC黑箱”，就是为我们的程序加入一些固化的‘C++代码’的东西”。

既然MFC黑箱帮我们加入了代码，那么大家想想它会帮我们加入什么样的代码呢？他会帮我们加入求解一元二次方程的代码吗？当然不会，所以它加入的实际上是每次编写窗口程序必须的，通用的代码。

再往下想，什么才是通用的呢？我们每次视窗编程都要写WinMain()函数，都要有注册窗口，产生窗口，消息循环，回调函数……即然每次都想要的东西，就让他们从我们眼前消失，让MFC帮忙写入！

手动模拟MFC程序的初始化

要知道MFC初始化过程，大家当然可以跟踪执行程序。但这种跟踪很麻烦，我相信大家都会跟踪的晕头转向。本人觉得哪怕你理解了MFC代码，也很容易让人找不着北，我们完全不懂的时候，在成千上万程序的迷宫中如何能找到出口？

我们要换一种方法，不如就来重新编写个MFC库吧，唉！大家不要笑，小心你的大牙，我不是疯子（虽然疯子也说自己不疯）。我们要写的就是最简单的MFC类库，就是把MFC宏观上的，理论上的东西写出来。我们要用最简化的代码，简化到刚好能运行。

1、需要“重写”的MFC库

既然，我们这一节写的是MFC程序的初始化过程，上面我们还有一个可执行的MFC程序。程序中只是用了两个MFC类，一个是CWinApp，另一个是CFrameWnd。当然，还有很多同样重要MFC类如视图类，文档类等等。但在上面的程序可以不用到，所以暂时省去了它（总之是为了简单）。

好，现在开始写MFC类库吧……唉，面前又有一个大难题，就是让大家背一下MFC层次结构图。天，那张鱼网怎么记得住，但既然我们要理解他，总得知道它是从那里派生出来的吧。

考虑到大家都很辛苦，那我们看一下上面两个类的父子关系（箭头代表派生）：

CObject->CCmdTarget->CWinThread->CWinApp->自己的重写了InitInstance()的应用程序类。
CObject（同上）->CCmdTarget（同上）->CWnd->CFrameWnd

看到层次关系图之后，终于可以开始写MFC类库了。按照上面层次结构，我们可以写以下六个类（为了直观，省去了构造函数和析构函数）。

c++代码

```
1.  ////////////////////////////////////////////
2.  class CObject{}; //MFC类的基类。
3.  class CCmdTarget : public CObject{};
4.  -----
5.  class CWinThread : public CCmdTarget{};
6.  class CWinApp : public CWinThread{};
7.  -----
8.  class CWnd : public CCmdTarget{};
```

新浪 (12)

微博 (11)

软件工程师 (10)

诺基亚 (10)

京东商城 (10)

比特币 (10)

Facebook (9)

周鸿祎 (9)

操作系统 (8)

Galaxy (8)

社交网络 (8)

搜索引擎 (8)

移动互联网 (8)

C (8)

亚马逊 (7)

更多标签

分享到

一键分享

QQ空间

新浪微博

百度云收藏

人人网

腾讯微博

百度相册

开心网

腾讯朋友

百度贴吧

豆瓣网

搜狐微博

百度新首页

QQ好友

和讯微博

更多...

百度分享

完全随机文章

VS2010/MFC

从Facebook和

VS2010/MFC

VS2010/MFC

VS2010/MFC

VS2010/MFC

程序员的選擇：技術vs管理

2015产品校招——阿里腾讯百度360小米...

C、C++、python、Java、php、C#六种流行...

App推广秘籍最全篇

程序员修炼指南——引导你成为真正的...

85后工作5年工资竟然涨了25倍——月薪...

TCP/UDP网络编程入门教程之十五：TC...

memcached使用场景和方法总结

从《奋斗》到《欢乐颂》看青年的价值观...

StackOverflow 创始人推荐程序员看...

魅族的掉队已成事实，生态链不是那么...

详解HTML5 LocalStorage本地存储

如何避免成为下一个雅虎

说说Javascript闭包这点事

最新评论及回复

CAddSheet(LPCTSTR ...

有没有XTP的教学？求教

就画个界面，搞这么负责，难怪MFC要被淘汰

讲的太好了，很全，很清楚！楼主你的Q...

楼主，请问如何动态给重写的CList...

普通人只有被剥削的份

已点广，，，，，告支持楼主

#include <afxco...

关掉王者荣耀。它就像鸦片，勾引小孩子...

关掉王者荣耀。它就像鸦片，勾引小孩子

楼主，请问CTabCtrl和CLis...

一、初始化函数中在设置好子对话框位置...

创建两组Radio可以在Radio的...

一定要通过【类向导】添加类，【类向导...

写的太棒了。

谢谢楼主

加油

弱的问一声？符号常量的用法是否跟C语...

蛮实用的可是在8年后才看到[REV...

好的程序员一定是挣钱的

最近发表

鸡啄米开始承接项目啦

```
9.     class CFrameWnd : public CWnd{};
10.    //////////////////////////////////////
```

大家再想一下，在上面的类里面，应该有什么？大家马上会想到，CWinApp类或者它的基本类CCmdTarget里面应该有一个虚函数virtual BOOL InitInstance()，是的，因为那里是程序的入口点，初始化程序的地方，那自然少不了的。可能有些朋友会说，反正InitInstance()在派生类中一定要重载，我不在CCmdTarget或CWinApp类里定义，留待CWinApp的派生类去增加这个函数可不可以。扯到这个问题可能有点越说越远，但我想信C++的朋友对虚函数应该是没有太多的问题的。总的来说，作为程序员如果清楚知道基类的某个函数要被派生类用到，那定义为虚函数要方便很多。

也有很多朋友问，C++为什么不自动把基类的所有函数定义为虚函数呢，这样可以省了很多麻烦，这样所有函数都遵照派生类有定义的函数就调用派生类的，没定义的就调用基类的，不用写virtual的麻烦多好！其实，很多面向对象的语言都这样做了。但定义一个虚函数要生成一个虚函数表，要占用系统空间，虚函数越多，表就越大，有时得不偿失！这里哆嗦几句，是因为往后要说明的消息映射中大家更加会体验到这一点，好了，就此打住。

上面我们自己解决了一个问题，就是在CCmdTarget写一个virtual BOOL InitInstance()。

2、WinMain()函数和CWinApp类

大家再往下想，我们还要我们MFC“隐藏”更多的东西：WinMain()函数，设计窗口类，窗口注册，消息循环，回调函数.....我们马上想到封装想封装他们。大家似乎隐约地感觉到封装WinMain()不容易，觉得WinMain()是一个特殊的函数，许多时候它代表了一个程序的起始和终结。所以在以前写程序的时候，我们写程序习惯从WinMain()的左大括写起，到右大括弧返回、结束程序。

我们换一个角度去想，有什么东西可以拿到WinMain()外面去做，许多初学者们，总觉得WinMain()函数是天大的函数，什么函数都好像要在它里面才能真正运行。其实这样了解很片面，甚至错误。我们可以写一个这样的C++程序：

```
c++代码
1.    //////////////////////////////////////
2.    #include <iostream.h>
3.    class test{
4.    public:
5.        test() {cout<<"请改变你对main() 函数的看法！"<<endl;}
6.    };
7.    test test1;
8.    /*****/
9.    void main() {}
10.   //////////////////////////////////////
```

在上面的程序里，入口的main()函数表面上什么也不做，但程序执行了（注：实际入口函数做了一些我们可以不了解的事情），并输出了一句话（注：全局对象比main()首先运行）。现在大家可以知道我们的WinMain()函数可以什么都不做，程序依然可以运行，但没有这个入口函数程序会报错。

那么WinMain()函数会放哪个类上面呢，请看下面程序：

```
c++代码
1.    #include <afxwin.h>
2.    class MyApp : public CWinApp
3.    {
4.    public:
5.        BOOL InitInstance()    //②程序入点
6.        {
7.            AfxMessageBox("程序依然可以运行！");
8.            return true;
9.        }
10.   };
11.   
```

小白照样读懂的VLAN原理讲解

SSH电商项目实战之十：商品类基本模块的搭建

SSH电商项目实战之九：添加和更新商品类别功能的实现

SSH电商项目实战之八：查询和删除商品类别功能的实现

SSH电商项目实战之七：Struts2和Json的整合

长文：内容产业的赢家与输家

SSH电商项目实战之六：基于DataGrid的数据显示

SSH电商项目实战之五：完成数据库的级联查询和分页

SSH电商项目实战之四：完成数据库的级联查询和分页

SSH电商项目实战之三：完成数据库的级联查询和分页

SSH电商项目实战之二：完成数据库的级联查询和分页

SSH电商项目实战之一：完成数据库的级联查询和分页

SSH电商项目实战之十：商品类基本模块的搭建

SSH电商项目实战之九：添加和更新商品类别功能的实现

SSH电商项目实战之八：查询和删除商品类别功能的实现

SSH电商项目实战之七：Struts2和Json的整合

SSH电商项目实战之六：基于DataGrid的数据显示

SSH电商项目实战之五：完成数据库的级联查询和分页

SSH电商项目实战之四：完成数据库的级联查询和分页

SSH电商项目实战之三：完成数据库的级联查询和分页

SSH电商项目实战之二：完成数据库的级联查询和分页

SSH电商项目实战之一：完成数据库的级联查询和分页

分享到

一键分享

QQ空间

新浪微博

百度云收藏

人人网

腾讯微博

百度相册

开心网

腾讯朋友

百度贴吧

豆瓣网

搜狐微博

百度新首页

QQ好友

和讯微博

更多...

百度分享

```
12. MyApp theApp; //①建立应用程序。
```

大家可以看到，我并没有构造框架，而程序却可以运行了——弹出一个对话框（如果没有WinMain()函数程序会报错）。上面我这样写还是为了直观起见，其实我们只要写两行程序：

```
#include <afxwin.h>
CWinApp theApp; //整个程序只构造一个CWinApp类对象，程序就可以运行！
```

所以说，只要我们构造了CWinApp对象，就可以执行WinMain()函数。我们马上相信WinMain()函数是在CWinApp类或它的基类中，而不是在其他类中。其实这种看法是错误的，我们知道编写C++程序的时候，不可能让你在一个类中包含入口函数，WinMain()是由系统调用，跟我们的平时程序自身调用的函数有着本质的区别。我们可以暂时简单想象成，当CWinApp对象构造完的时候，WinMain()跟着执行。

现在大家明白了，大部分的“通用代码（我们想封装隐藏的东西）”都可以放到CWinApp类中，那么它又是怎样运行起来的呢？为什么构造了CWinApp类对象就“自动”执行那么多东西。

大家再仔细想一下，CWinApp类对象构造之后，它会“自动”执行自己的构造函数。那么我们可以把想要“自动”执行的代码放到CWinApp类的构造函数中。

那么CWinApp类可能打算这样设计（先不计较正确与否）：

C++代码

```
1. class CWinApp : public CWinThead{
2. public:
3. virtual BOOL InitInstance(); //解释过的程序的入口
4. CWinApp::CWinApp() { //构造函数
5. //////////////////////////////////////////////////
6. WinMain(); //这个是大家一眼看出的错误
7. Create(); //设计、创建、更新显示窗口
8. Run(); //消息循环
9. //////////////////////////////////////////////////
10. }
11. };
```

写完后，大家又马上感觉到似乎不对，WinMain()函数在这里好象真的一点用处都没有，并且能这样被调用吗（请允许我把手按在圣经上声明一下：WinMain()不是普通的函数，它要肩负着初始化应用程序，包括全局变量的初始化，是由系统而不是程序本身调用的，WinMain()返回之后，程序就结束了，进程撤消）。再看Create()函数，它能确定设计什么样的窗口，创建什么样的窗口吗？如果能在CWinApp的构造函数里确定的话，我们以后设计MFC程序时窗口就一个样，这样似乎不太合理。

回过头来，我们可以让WinMain()函数一条语句都不包含吗？不可以，我们看一下WinMain()函数的四个参数：

```
WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
```

其中第一个参数指向一个实例句柄，我们在设计WNDCLASS的时候一定要指定实例句柄。我们窗口编程，肯定要设计窗口类。所以，WinMain()再简单也要这样写：

```
int WinMain(HINSTANCE hinst, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{ hInstance=hinst }
```

既然实例句柄要等到程序开始执行才能知道，那么我们用于创建窗口的Create()函数也要在WinMain()内部才能执行（因为如果等到WinMain()执行完毕后，程序结束，进程撤消，当然Create()也不可能创建窗口）。

再看Run()（消息循环）函数，它能在WinMain()函数外面运行吗？众所周知，消息循环就是相同的那么几句代码，但我们也不要企图把它放在WinMain()函数之外执行。

分享到

一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

百度分享

分享

所以我们的WinMain()函数可以像下面这样写：

```
WinMain(.....)
{
    .....窗口类对象执行创建窗口函数.....
    .....程序类对象执行消息循环函数.....
}
```

对于WinMain()的问题，得总结一下，我们封装的时候是不可以把它封装到CWinApp类里面，但由于WinMain()的不变性（或者说有规律可循），MFC完全有能力在我们构造CWinApp类对象的时候，帮我们完成那几行代码。

转了一个大圈，我们仿佛又回到了SDK编程的开始。但现在我们现在能清楚地知道，表面上MFC与SDK编程截然不同，但实质上MFC只是用类的形式封装了SDK函数，封装之后，我们在WinMain()函数中只需要几行代码，就可以完成一个窗口程序。我们也由此知道了应如何去封装应用程序类（CWinApp）和主框架窗口类（CFrameWnd）。下面把上开始设计这两个类。

3、MFC库的“重写”

为了简单起见，我们忽略这两个类的基类和派生类的编写，可能大家会认为这是一种很不负责任的做法，但本人觉得这既可减轻负担，又免了大家在各类之间穿来穿去，更好理解一些（我们在关键的地方作注明）。还有，我把全部代码写在同一个文件中，让大家看起来不用那么吃力，但这是最不提倡的写代码方法，大家不要学哦！

分享到

一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

百度分享

分享

c++代码

```
1.  #include <windows.h>
2.  HINSTANCE hInstance;
3.
4.  class CFrameWnd
5.  {
6.      HWND hwnd;
7.  public:
8.      CFrameWnd();    //也可以在这里调用Create()
9.      virtual ~CFrameWnd();
10.     int Create();    //类就留意这一个函数就行了!
11.     BOOL ShowWnd();
12. };
13. class CWinApp1
14. {
15. public:
16.     CFrameWnd* m_pMainWnd; //在真正的MFC里面
17.     //它是CWnd指针，但这里由于不写CWnd类
18.     //只要把它写成CFrameWnd指针
19.     CWinApp1* m_pCurrentWinApp; //指向应用程序对象本身
20.     CWinApp1();
21.     virtual ~CWinApp1();
22.     virtual BOOL InitInstance(); //MFC原本是必须重载的函数，最重要的函数!!!
23.
24.     virtual BOOL Run(); //消息循环
25. };
26. CFrameWnd::CFrameWnd() {}
27. CFrameWnd::~CFrameWnd() {}
28.
29. int CFrameWnd::Create()    //封装创建窗口代码
30. {
31.     WNDCLASS wndcls;
32.     wndcls.style=0;
```



```
33.   wndcls.cbWndExtra=0;
34.   wndcls.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
35.   wndcls.hCursor=LoadCursor(NULL, IDC_CROSS);
36.   wndcls.hIcon=LoadIcon(NULL, IDC_ARROW);
37.   wndcls.hInstance=hInstance;
38.   wndcls.lpfnWndProc=DefWindowProc; //默认窗口过程函数。
39.   //大家可以想象成MFC通用的窗口过程。
40.   wndcls.lpszClassName="窗口类名";
41.   wndcls.lpszMenuName=NULL;
42.   RegisterClass(&wndcls);
43.
44.   hwnd=CreateWindow("窗口类名", "窗口实例标题名", WS_OVERLAPPEDWINDOW, 0, 0, 60
0, 400, NULL, NULL, hInstance, NULL);
45.   return 0;
46. }
47.
48. BOOL CFrameWnd::ShowWnd() //显示更新窗口
49. {
50.   ShowWindow(hwnd, SW_SHOWNORMAL);
51.   UpdateWindow(hwnd);
52.   return 0;
53. }
54.
55. ///////////////
56. CWinApp1::CWinApp1()
57. {
58.   m_pCurrentWinApp=this;
59. }
60. CWinApp1::~CWinApp1() {}
61. //以下为InitInstance()函数，MFC中要为CWinApp的派生类改写，
62. //这里为了方便理解，把它放在CWinApp类里面完成！
63. //你只要记住真正的MFC在派生类改写此函数就行了。
64. BOOL CWinApp1::InitInstance()
65. {
66.   m_pMainWnd=new CFrameWnd;
67.   m_pMainWnd->Create();
68.   m_pMainWnd->ShowWnd();
69.   return 0;
70. }
71.
72. BOOL CWinApp1::Run() ////////////////封装消息循环
73. {
74.   MSG msg;
75.   while(GetMessage(&msg, NULL, 0, 0))
76.   {
77.     TranslateMessage(&msg);
78.     DispatchMessage(&msg);
79.   }
80.   return 0;
81. } ////////////////封装消息循环
82.
83. CWinApp1 theApp; //应用程序对象（全局）
84.
85. int WINAPI WinMain( HINSTANCE hinst, HINSTANCE hPrevInstance, LPST
R lpCmdLine, int nCmdShow)
```

分享到

一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

百度分享

```
86. {
87.     hInstance=hinst;
88.     CWinApp* pApp=theApp.m_pCurrentWinApp;
89.     //真正的MFC要写一个全局函数AfxGetApp, 以获取CWinApp指针。
90.     pApp->InitInstance();
91.     pApp->Run();
92.     return 0;
93. }
```

代码那么长, 实际上只是写了三个函数, 一是CFrameWnd类的Create(), 第二个是CWinApp类的InitInstance()和Run()。在此特别要说明的是InitInstance(), 真正的MFC中, 那是我们跟据自己构造窗口的需要, 自己改写这个函数。

大家可以看到, 封装了上面两个类以后, 在入口函数WinMain中就写几行代码, 就可以产生一个窗口程序。在MFC中, 因为WinMain函数就是固定的那么几行代码, 所以MFC绝对可以帮助我们自动完成 (MFC的特长就是帮我们完成有规律的代码), 也因此我们创建MFC应用程序的时候, 看不到WinMain函数。

分享源自: <http://blog.csdn.net/liyi268/article/details/297875>

除非特别注明, 鸡啄米文章均为原创

转载请标明本文地址: <http://www.jizhuomi.com/software/267.html>

2012年11月22日

作者:鸡啄米 分类:软件开发 浏览:250859 评论:12

相关文章:

[VS2010中如何实现自定义MFC控件](#) (2012-11-18 22:56:16)

[VS2010/MFC编程入门教程之目录和总结](#) (2012-10-31 22:4:12)

[VS2010/MFC编程入门之五十四 \(Ribbon界面开发: 使用更多控件并为控件添加消息处理函数\)](#) (2012-10-27 21:56:47)

[VS2010/MFC编程入门之五十三 \(Ribbon界面开发: 为Ribbon Bar添加控件\)](#) (2012-10-21 21:53:16)

[VS2010/MFC编程入门之五十二 \(Ribbon界面开发: 创建Ribbon样式的应用程序框架\)](#) (2012-10-17 22:0:34)

[VS2010/MFC编程入门之五十一 \(图形图像: GDI对象之画刷CBrush\)](#) (2012-10-12 22:19:2)

[VS2010/MFC编程入门之五十 \(图形图像: GDI对象之画笔CPen\)](#) (2012-10-8 22:53:50)

[VS2010/MFC编程入门之四十九 \(图形图像: CDC类及其屏幕绘图函数\)](#) (2012-9-26 21:22:28)

[VS2010/MFC编程入门之四十八 \(字体和文本输出: 文本输出\)](#) (2012-9-22 22:41:48)

[VS2010/MFC编程入门之四十七 \(字体和文本输出: CFont字体类\)](#) (2012-9-18 22:53:35)

1楼. 医院标语

路过, 过来看看。欢迎不?

鸡啄米 于 2012-11-23 23:20:06 回复
必须欢迎啊

2012/11/23 14:09:45 [回复该留言](#)

2楼. 吴克难的博客

学到了不少知识, 感谢博主的分享~

2012/11/23 15:01:25 [回复该留言](#)

3楼. 郭涵倩

楼主好厉害啊, 学习下

鸡啄米 于 2012-11-25 22:41:32 回复
过奖了, 常来啊

分享到

一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

百度分享

分享

2012/11/23 17:19:26 [回复该留言](#)

4楼. 祝北平

专业博客 缔造专业

鸡啄米 于 2012-11-25 22:42:09 回复
言简意赅

2012/11/24 12:16:46 [回复该留言](#)

5楼. 润初颜

技术牛人！

2012/11/24 13:47:27 [回复该留言](#)

6楼. easy

请问如何链接MFC库？谢谢

2013/11/11 14:15:35 [回复该留言](#)

7楼. 简单

我复制了后面的代码，为什么运行不了，显示有错误。
作者能否详细说下，如果要运行该程序，还需要哪些设置？

2013/11/12 12:39:01 [回复该留言](#)

8楼. 简单

analogMFC_2.obj : error LNK2019: 无法解析的外部符号 __imp__CreateWindowExA@48，该符号在函数 "public: int __thiscall CFrameWnd::Create(void)" (?Create@CFrameWnd@@@QAEHXZ) 中被引用
1>analogMFC_2.obj : error LNK2019: 无法解析的外部符号 __imp__RegisterClassA@4，该符号在函数 "public: int __thiscall CFrameWnd::Create(void)" (?Create@CFrameWnd@@@QAEHXZ) 中被引用
1>analogMFC_2.obj : error LNK2019: 无法解析的外部符号 __imp__DefWindowProcA@16，该符号在函数 "public: int __thiscall CFrameWnd::Create(void)" (?Create@CFrameWnd@@@QAEHXZ) 中被引用
1>analogMFC_2.obj : error LNK2019: 无法解析的外部符号 __imp__LoadIconA@8，该符号在函数 "public: int __thiscall CFrameWnd::Create(void)" (?Create@CFrameWnd@@@QAEHXZ) 中被引用
1>analogMFC_2.obj : error LNK2019: 无法解析的外部符号 __imp__LoadCursorA@8，该符号在函数 "public: int __thiscall CFrameWnd::Create(void)" (?Create@CFrameWnd@@@QAEHXZ) 中被引用
显示如下错误！

2013/11/12 12:39:46 [回复该留言](#)

9楼. ssspk

后面的代码不要链接MFC库就可以用了。感谢博主的分享！

2013/12/26 16:04:35 [回复该留言](#)

10楼. 学无止境

"我们岂能忍受这种"黑盒"操作？" 说的好，开发人员就应该具有这样的想法，刨根问底。

2014/4/23 15:44:11 [回复该留言](#)

11楼. 新手

为什么我输入那个空的main函数编译，提示无法启动程序？
#include <iostream>
using namespace std;
class test
{
public:
test()
{
cout << "请改变你对main () 函数的看法!" << endl;
}
};
test test1;

分享到

一键分享	QQ空间
新浪微博	百度云收藏
人人网	腾讯微博
百度相册	开心网
腾讯朋友	百度贴吧
豆瓣网	搜狐微博
百度新首页	QQ好友
和讯微博	更多...

[百度分享](#)

分享


```
/*******/
```

```
void main()
{

}
```

```
////////
```

2015/6/22 18:39:09 [回复该留言](#)

12楼 [crissw2000](#)


能够看到我吗？如果看到，请回复，有一些问题想请教。谢谢！

2018/1/24 10:58:02 [回复该留言](#)

上一篇：[草根创业者能否在移动互联网中缔造传奇](#)

下一篇：[解密为何iOS设备较之Android设备更加流畅](#)

发表评论:

<input type="text"/>	名称(*)
<input type="text"/>	邮箱(选填)
<input type="text"/>	网站链接(选填)
<input type="text"/>	验证(*) 

正文(*) (留言最长字数:1000)

☐ 记住我,下次回复时不用重新输入个人信息

[\[URL\]](#) [\[URL2\]](#) [\[EMAIL\]](#) [\[EMAIL2\]](#) [\[B\]](#) [\[I\]](#) [\[U\]](#) [\[S\]](#) [\[QUOTE\]](#) [显示UBB表情>>](#)

◎欢迎参与讨论，请在这里发表您的看法、交流您的观点。

分享到

- | | |
|-------|-------|
| 一键分享 | QQ空间 |
| 新浪微博 | 百度云收藏 |
| 人人网 | 腾讯微博 |
| 百度相册 | 开心网 |
| 腾讯朋友 | 百度贴吧 |
| 豆瓣网 | 搜狐微博 |
| 百度新首页 | QQ好友 |
| 和讯微博 | 更多... |

[百度分享](#)

分享