

# 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六大核心机制之四：永久保存（串行化）

## MFC六大核心机制之四：永久保存（串行化）

分类标签: [MFC](#) [VC++](#)

永久保存（串行化）是MFC的重要内容，可以用一句简明直白的话来形容其重要性：弄懂它以后，你就越来越像个程序员了！

如果我们的程序不需要永久保存，那几乎可以肯定是一个小玩儿。那怕我们的记事本、画图等小程序，也需要保存才有真正的意义。

对于MFC的很多地方我不甚满意，总觉得它喜欢拿一组低能而神秘的宏来故弄玄虚，但对于它的连续存储（serialize）机制，却是我十分钟爱的地方。在此，可让大家感受到面向对象的幸福。

MFC的连续存储（serialize）机制俗称串行化。“在你的程序中尽管有着各种各样的数据，serialize机制会象流水一样按顺序存储到单一的文件中，而又能按顺序地取出，变成各种不同的对象数据。”不知我在说上面这一句话的时候，大家有什么反应，可能很多朋友直觉是一件很简单的事情，只是说了一个“爽”字就没有下文了。

### 串行化原理的讨论

要实现象流水一样存储其实是一个很大的难题。试想，在我们的程序里有各式各样的对象数据。如画图程序中，里面设计了点类，矩形类，圆形类等等，它们的绘图方式及对数据的处理各不相同，用它们实现了成百上千的对象之后，如何存储起来？不想由可，一想头都大了：我们要在程序中设计函数store()，在我们单击“文件/保存”时能把各对象往里存储。那么这个store()函数要神通广大，它能清楚地知道我们设计的是什么样的类，产生什么样的对象。大家可能并不觉得这是一件很困难的事情，程序有能力知道我们的类的样子，对象也不过是一块初始化了存储区域罢了。就把一大堆对象“转换”成磁盘文件就行了。

即使上面的存储能成立，但当我们单击“文件/打开”时，程序当然不能预测用户想打开哪个文件，并且当打开文件的时候，要根据你那一堆垃圾数据new出数百个对象，还原为你原来存储时的样子，你又该怎么做呢？

试想，要是我们有一个能容纳各种不同对象的容器，这样，用户用我们的应用程序打开一个磁盘文件时，就可以把文件的内容读进我们程序的容器中。把磁盘文件读进内存，然后识别它“是什么对象”是一件很难的事情。首先，保存过程不像电影的胶片，把景物直接映射进去，然后，看一下胶片就知道那是什么内容。可能有朋友说它象录像磁带，拿着录像带我们看不出里面变化的磁场信号，但经过录像机就能把它还原出来。

其实不是这样的，比如保存一个矩形，程序并不是把矩形本身按点阵存储到磁盘中，因为我们绘制矩形的整个过程只不过是调用一个GDI函数罢了。它保存只是坐标值、线宽和某些标记等。程序面对“00 FF”这样的东西，当然不知道它是一个圆或是一个字符！

拿刚才录像带的例子，我们之所以能最后放映出来，前提我们知道这对象是“录像带”，即确定了它是什么类对象。如果我们事先只知道它“里面保存有东西，但不知道它是什么类型的东西”，这就导致我们无法把它读出来。拿录像带到录音机去放，对录音机来说，那完全是垃圾数据。即是说，要了解永久保存，要对动态创建有深刻的认识。

现在大家可以知道困难的根源了吧。我们在写程序的时候，会不断创造新的类，构造新的对象。这些对象，当然是旧的类对象（如MyDocument）从未见过的。那么，我们如何才能使文档对象可以保存自己新对象呢，又能动态创建自己新的类对象呢？

许多朋友在这个时候想起了CObject这个类，也想到了虚函数的概念。于是以为自己“大致了解”串行化的概念。他们设想：“我们设计的MyClass（我们想用于串行化的对象）全部从CObject类派生，

订阅鸡啄米

[RSS](#) [+ 订阅到QQ邮箱](#)

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 \(12\)](#)

CObject类对象当然是MyDocument能认识的。”这样就实现了一个目的：本来MyDocument不能识别我们创建的MyClass对象，但它能识别CObject类对象。由于MyClass从CObject类派生，构造的新类对象“是一个CObject”，所以MyDocument能把我们的新对象当作CObject对象读出。或者根据书本上所说的：打开或保存文件的时候，MyDocument会调用Serialize()，MyDocument的Serialize()函数会呼叫我们创建类的Serialize函数[即是在MyDocument Serialize()中调用：m\_pObject->Serialize()，注意：在此m\_pObject是CObject类指针，它可以指向我们设计的类对象]。最终结果是MyDocument的读出和保存变成了我们创建的类对象的读出和保存，这种认识是不明朗的。

有意思还有，在网上我遇到几位自以为懂了Serialize的朋友，居然不约而同的犯了一个很低级得让人不可思议的错误。他们说：Serialize太简单了！Serialize()是一个**虚函数**，虚函数的作用就是“优先派生类的操作”。所以MyDocument不实现Serialize()函数，留给我们自己的MyClass对象去调用Serialize().....真是哭笑不得，我们创建的类MyClass并不是由MyDocument类派生，Serialize()函数为虚在MyDocument和MyClass之间没有任何意义。MyClass产生的MyObject对象仅仅是MyDocument的一个成员变量罢了。

话说回来，由于MyClass从CObject派生，所以CObject类型指针能指向MyClass对象，并且能够让MyClass对象执行某些函数（特指重载的CObject虚函数），但前提必须在MyClass对象实例化了，即在内存中占领了一块存储区域之后。不过，我们的问题恰恰就是在应用程序随便打开一个文件，面对的是它不认识的MyClass类，当然实例化不了对象。

幸好我们在上一节课中懂得了**动态创建**。即想要从CObject派生的MyClass成为可以动态创建的对象只要用到DECLARE\_DYNAMIC/IMPLEMENT\_DYNAMIC宏就可以了（注意：最终可以Serialize的对象仅仅用到了DECLARE\_SERIAL/IMPLEMENT\_SERIAL宏，这是因为DECLARE\_SERIAL/IMPLEMENT\_SERIAL包含了DECLARE\_DYNAMIC/IMPLEMENT\_DYNAMIC宏）。

### 整理思路，深入理解串行化

从解决上面的问题中，我们可以分步理解了：

- 1、Serialize的目的：让MyDocument对象在执行打开/保存操作时，能读出（构造）和保存它不认识的MyClass类对象。
- 2、MyDocument对象在执行打开/保存操作时会调用它本身的Serialize()函数。但不要指望它会自动保存和读出我们的MyClass类对象。这个问题很容易解决，如下即可：

#### c++代码

```
1. MyDocument:: Serialize () {
2.     // 在此函数调用MyClass类的Serialize () 就行了！即
3.     MyObject. Serialize () ;
4. }
```

- 3、我们希望MyClass对象为可以动态创建的对象，所以要求在MyClass类中加上DECLARE\_DYNAMIC/IMPLEMENT\_DYNAMIC宏。

但目前的Serialize机制还很抽象。我们仅仅知道了表面上的东西，实际又是如何的呢？下面作一个简单深刻的详解。

先看一下我们文档类的Serialize()：

#### c++代码

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

### 完全随机文章

VS2010/MFC编程入门之五（MFC消息映...  
从Facebook和Twitter获取新闻的用户...  
VS2010/MFC编程入门之二十五（常用控...  
VS2010/MFC编程入门之三十五（菜单：...  
VS2010/MFC编程入门之四十一（文档、...  
VS2010/MFC编程入门教程之目录和总...  
程序员的选择：技术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

```
1. void CMyDoc::Serialize(CArchive& ar)
2. {
3.     if (ar.IsStoring())
4.     {
5.         // TODO: add storing code here
6.     }
7.     else
8.     {
9.         // TODO: add loading code here
10.    }
11. }
```

目前这个子数什么也没做（没有数据的读出和写入），CMyDoc类正等待着我们去改写这个函数。现在假设CMyDoc有一个MFC可识别的成员变量m\_MyVar，那么函数就可改写成如下形式：

#### c++代码

```
1. void CMyDoc::Serialize(CArchive& ar)
2. {
3.     if (ar.IsStoring()) //读写判断
4.     {
5.         ar<<m_MyVar; //写
6.     }
7.     else
8.     {
9.         ar>>m_MyVar; //读
10.    }
11. }
```

许多网友问：自己写的类（即MFC未包含的类）为什么不行？我们在CMyDoc里包含自写类的头文件MyClass.h，这样CMyDoc就认识MyDoc类对象了。这是一般常识性的错误，MyDoc类认识MyClass类对象与否并没有用，关键是CArchive类，即对象ar不认识MyClass（当然你梦想重写CArchive类当别论）。“>>”、“<<”都是CArchive重载的操作符。上面ar>>m\_MyVar说白即是在执行一个以ar和m\_MyVar为参数的函数，类似于function(ar,m\_MyVar)罢了。我们当然不能传递一个它不认识的参数类型，也因此不会执行function(ar,m\_MyObject)了。

【注：这里我们可以用指针。让MyClass从CObject派生，一切又起了质的变化，假设我们定义了：MyClass \*pMyClass = new MyClass;因为MyClass从CObject派生，根据虚函数原理，pMyClass也是一个CObject\*，即pMyClass指针是CArchive类可认识的。所以执行上述function(ar, pMyClass)，即ar << pMyClass是没有太多的问题（在保证MyClass对象可以动态创建的前提下）。】

回过头来，如果想让MyClass类对象能Serialize，就得让MyClass从CObject派生，Serialize()函数在CObject里为虚，MyClass从CObject派生之后就可以根据自己的要求去改写它，像上面改写CMyDoc::Serialize()方法一样。这样MyClass就得到了属于MyClass自己特有的Serialize()函数。

现在，程序就可以这样写：

#### c++代码

```
1. ....
2.
3. #include "MyClass.h"
4.
5. ....
6.
7. void CMyDoc::Serialize(CArchive& ar)
8. {
9.     //在此调用MyClass重写过的Serialize()
10.    m_MyObject.Serialize(ar); // m_MyObject为MyClass实例
```

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：EasyUI菜单的实现

SSH电商项目实战之三：使用EasyUI搭建后台页面框架

SSH电商项目实战之二：基本增删查改、Service和Action的抽取以及使用注解替换xml

大妈：我们不懂ICO和X币，但知道比炒房厉害

SSH电商项目实战之一：整合Struts2、Hibernate和Spring

面临连续亏损，HTC出售手机还是VR业务？

分享

```
11. }
```

至此，串行化工作就算完成了，简单直观的讲：从CObject派生自己的类，重写Serialize()。在此过程中，我刻意安排：在没有用到DECLARE\_SERIAL/IMPLEMENT\_SERIAL宏，也没有用到CArray等模板类的前提下就完成了串行化的工作。我看过某些书，总是一开始就讲DECLARE\_SERIAL/IMPLEMENT\_SERIAL宏或马上用CArray模板，让读者觉得串行化就是这两个东西，导致许多朋友因此找不着北。

大家看到了，没有DECLARE\_SERIAL/IMPLEMENT\_SERIAL宏和CArray等数据结构模板也依然可以完成串行化工作。

## CArchive

最后再补充讲解一下有些抽象的CArchive。我们先看以下程序（注：以下程序包含动态创建等，请包含DECLARE\_SERIAL/IMPLEMENT\_SERIAL宏）

c++代码

```
1. void MyClass::Serialize(CArchive& ar)
2. {
3.     if (ar.IsStoring()) //读写判断
4.     {
5.         ar<< m_pMyVar; //问题: ar 如何把m_pMyVar所指的对象变量保存到磁
        盘?
6.     }
7.     else
8.     {
9.         pMyClass = new MyClass; //准备存储空间
10.        ar>> m_pMyVar;
11.    }
12. }
```

为回答上面的问题，即“ar<<XXX”的问题，这里给出一段模拟CArchive的代码。

“ar<<XXX”是执行CArchive对运算符“<<”的重载动作。ar和XXX都是该重载函数中的一参数而已。函数大致如下：

c++代码

```
1. CArchive& operator<<( CArchive& ar, const CObject* pObj)
2. {
3.     .....
4.     //以下为CRuntimeClass链表中找到、识别pObj资料。
5.     CRuntimeClass* pClassRef = pObj->GetRuntimeClass();
6.     //保存pClassRef即类信息（略）
7.
8.     ((CObject*)pObj)->Serialize(); //保存MyClass数据
9.     .....
10. }
```

从上面可以看出，因为Serialize()为虚函数，即“ar<<XXX”的结果是执行了XXX所指向对象本身的Serialize()。对于“ar>>XXX”，虽然不是“ar<<XXX”逆过程，大家可能根据动态创建和虚函数的原理料想到它。

至此，永久保存算是写完了。在此过程中，我一直努力用最少的代码，详尽的解释来说明问题。以前我为本课题写过一版本，并在几个论坛上发表过，但不知怎么在网上遗失（可能被删除）。所以这篇文章是我重写的版本。记得第一个版本中，我是对DECLARE\_SERIAL/IMPLEMENT\_SERIAL和可串行化的数组及链表对象说了许多。这个版本中我对DECLARE\_SERIAL/IMPLEMENT\_SERIAL其中奥秘几乎一句不提，目的是让大家能找到中心，有更简洁的永久保存的概念，我觉得这种感觉很好！

分享

来自: <http://blog.csdn.net/liyi268/article/details/623367>

除非特别说明，鸡啄米文章均为原创

转载请标明本文地址: <http://www.jizhuomi.com/software/273.html>

2012年12月4日

作者:鸡啄米 分类:软件开发 浏览:235214 评论:9

相关文章:

[MFC六大核心机制之三：动态创建](#) (2012-11-30 21:43:21)

[MFC六大核心机制之二：运行时类型识别 \(RTTI\)](#) (2012-11-26 21:5:33)

[MFC六大核心机制之一：MFC程序的初始化](#) (2012-11-22 22:20:43)

[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)

分享

#### 1楼. ixwebhosting

好专业的分享，互相学习一下

2012/12/5 8:56:58 [回复该留言](#)

#### 2楼. 旅途者

这么长啊，我真的没看完.....

鸡啄米 于 2012-12-05 23:11:55 回复  
呵呵，讲原理一般都太短不了

2012/12/5 15:03:46 [回复该留言](#)

#### 3楼. 润初颜

好专业！

2012/12/6 11:07:22 [回复该留言](#)

#### 4楼. 足球比分

想学习一系列的建站作战过程，就是不知道怎么下手，博主教我一下吧

2012/12/6 14:57:53 [回复该留言](#)

#### 5楼. siyue的小栈

郁闷，编程荒废很久了，看不明白了都。。。

2012/12/6 23:42:57 [回复该留言](#)

#### 6楼. Topub.me

MFC听说难。。还没接触。。

鸡啄米 于 2012-12-09 00:13:07 回复  
其实不是想象的那么难

2012/12/7 8:41:22 [回复该留言](#)

#### 7楼. dc

版主.能不能用VS2010基于对话框程序写一两个实现串行化例子出来？

比如实现不同类型的新建、存储、调出、修改、再存储等等系列操作。  
光看你上面的描述，现在还很难体现出来；只要有一两实例亲自动手去做过才能吸取个中乐趣。这样能会



使用。我很期待你的例子分享...

2012/12/11 15:44:04 [回复该留言](#)

8楼. dc

是了，版主有没有创建的一个VC QQ群啊？  
有条件的话建立个VC QQ群让我们进去跟你学习学习一下...因为在这里提问和交流还是不够实时和互动性



不好。

鸡啄米 于 2012-12-12 23:36:06 回复  
关于串行化，后面会考虑讲解。  
qq群目前已满，近期暂停建群，很抱歉。  
谢谢你的关注和支持。

2012/12/11 15:52:41 [回复该留言](#)

9楼. 菜鸟


好赞

2015/9/13 13:35:15 [回复该留言](#)

上一篇：[移动大势下昔日巨头英特尔路在何方](#)

下一篇：[苹果惧怕三星的理由](#)

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。