Atitit [持久](https://baike.baidu.com/item/%E6%8C%81%E4%B9%85" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)化 Persistence概念的艺术

目录

[1. 持久化是将程序数据在持久状态和瞬时状态间转换的机制。 1](#_Toc19655)

[2. DBC就是一种持久化机制。文件IO也是一种持久化机制。 2](#_Toc30567)

[3. 日常持久化的方法 2](#_Toc5918)

[4. 理解与分类 3](#_Toc25961)

[4.1. 二个层面 3](#_Toc3495)

[4.1.1. 应用层 3](#_Toc18749)

[4.1.2. 系统层 4](#_Toc28794)

[5. 持久化是一种对象服务实现至少3个接口 4](#_Toc18428)

[6. 类似概念序列化 4](#_Toc2107)

[7. 持久化方案可以分为关系数据库方案、文件方案、对象数据库方案、xml数据库方案 4](#_Toc10206)

# 持久化是将程序数据在持久状态和瞬时状态间转换的机制。

持久化 编辑

持久化是将程序数据在持久状态和瞬时状态间转换的机制。通俗的讲，就是瞬时数据（比如内存中的数据，是不能永久保存的）持久化为持久数据（比如持久化至数据库中，能够长久保存）。

中文名 持久化 外文名 Persistence 方 法 通过逻辑处理接口储存数据 理 解 持久化是针对时间来说的

目录

1 定义

2 理解

3 二个层面

▪ 应用层

▪ 系统层

4 特点

▪ 对象

▪ 市场

▪ 序列化

5 意义

▪ 兴起原因

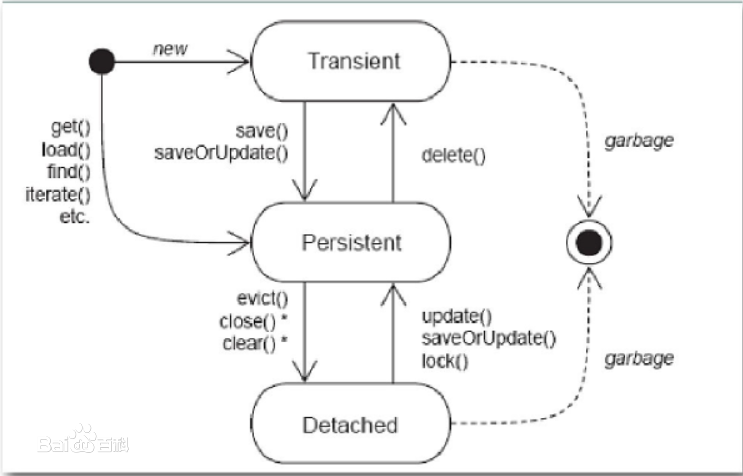
▪ 运用

6 相关实现

定义编辑

持久化（Persistence），即把数据（如内存中的对象）保存到可永久保存的存储设备中（如磁盘）。持久化的主要应用是将内存中的对象存储在数据库中，或者存储在磁盘文件中、XML数据文件中等等。

持久化是将程序数据在持久状态和瞬时状态间转换的机制。



# DBC就是一种持久化机制。文件IO也是一种持久化机制。

# 日常持久化的方法

将鲜肉冷藏，吃的时候再解冻的方法也是。

将水果做成罐头的方法也是。

将人的脏器迅速冷冻，运输，然后解冻给人移植的技术也是。

# 理解与分类

[编辑](https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/javascript:;)

我们这样理解：

在一定周期内保持不变就是持久化,持久化是针对时间来说的。

数据库中的数据就是持久化了的数据,只要你不去删除或修改。

比如在[IE浏览器](https://baike.baidu.com/item/IE%E6%B5%8F%E8%A7%88%E5%99%A8" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)中一次Session会话中Session对象变量也是不变的,是Session容器中持久化。

[对象持久化](https://baike.baidu.com/item/%E5%AF%B9%E8%B1%A1%E6%8C%81%E4%B9%85%E5%8C%96" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)的方式有很多种,根据周期不同有,page,Session,Application。

对象序列化机制对于需要将对象的状态保存到文件中，而后能够通过读入对象状态来重新构造对象，恢复程序状态,

对象序列化的过程是对象持久化的方法之一,把对象保存到文件中。

## 二个层面

[编辑](https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/javascript:;)

简单的理解持久化可以在二个层面：[应用层](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E5%B1%82" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)和系统层、

### 应用层

如果关闭(shutdown)你的应用然后重新启动则先前的数据依然存在。

### 系统层

如果关闭(shutdown)你的系统（电脑）然后重新启动则先前的数据依然存在。

# 持久化是一种对象服务实现至少3个接口

，就是把内存中的对象保存到外存中，让以后能够取回。需要实现至少3个接口：

void Save(object o) 把一个对象保存到外存中

Object Load(object oid) 通过对象标识从外存中取回对象

bool[Exists](https://baike.baidu.com/item/Exists" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)(object oid) 检查外存中是否存在某个对象.

# 类似概念序列化

我们先跳开一下，看看另一个类似的有用概念：序列化[Serialize](https://baike.baidu.com/item/Serialize" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)也是一种对象服务，就是把内存中的对象序列化成流、或者把流反序列化成对象。需要实现2个接口：

[void](https://baike.baidu.com/item/void" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank) [Serialize](https://baike.baidu.com/item/Serialize" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)([Stream](https://baike.baidu.com/item/Stream" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank) [stream](https://baike.baidu.com/item/stream" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank),object o) 把对象序列化到流中

[object](https://baike.baidu.com/item/object" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank) Deserialize(Stream stream) 把流反序列化成对象

序列化和持久化很相似，有些人甚至混为一谈，其实还是有区别的，序列化是为了解决对象的传输问题，传输可以在[线程](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)之间、进程之间、内存外存之间、[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)之间进行。我之所以在这里提到序列化，是因为我们可以利用序列化来辅助持久化，可以说凡是可以持久化的对象都可以序列化，因为序列化相对容易一些（也不是很容易），所以主流的软件基础设施，比如.net和java，已经把序列化的框架完成了。

# 持久化方案可以分为关系数据库方案、文件方案、对象数据库方案、[xml数据库](https://baike.baidu.com/item/xml%E6%95%B0%E6%8D%AE%E5%BA%93" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)方案

，现今主流的持久化方案是关系数据库方案，

关系数据库方案不仅解决了并发的问题，更重要的是，关系数据库还提供了持久化服务之外的价值：统计分析功能。刚才我说到，凡是可以序列化的对象都可以持久化，极端的说，我们可以只建立一个表Object(OID,Bytes)，但基本上没有人这么做，因为一旦这样，我们就失去了关系数据库额外的统计分析功能。

关系数据库和[面向对象](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/%E6%8C%81%E4%B9%85%E5%8C%96/_blank)之间有一条鸿沟，因为二者模式不匹配，所以就存在一个OR映射问题。