

# Module05-06

## C++ Boost: 序列化

- 容器相关
- 字符串和文字处理
- 正则表达式
- 智能指针
- 函数对象相关
- ➔ 序列化
- 日期与时间
- 多线程
- 网络

- 序列化：
  - ◆ 关于 serialization（序列化）
  - ◆ 使用 boost.serialization

## ■ 关于序列化

- ◆ 这里，我们用术语 "serialization" 来表示将任意一组 C++ 数据结构解构为一串字节的、可逆的过程。这样的系统可用于在另一个程序上下文中重新构建一个等价的结构。根据不同的上下文，它可以用来实现对象持久化、远程参数传递或其它功能。在本系统中，我们使用术语 "archive 存档" 来指代这个字节流的特定表现。它可以是一个二进制数据文件、文本文件、XML 或其它由本库的用户所创建的东西。

- 关于 boost.serialization 的实现目标（能力）
  - ◆ 代码的可移植性 - 只依赖于 ANSI C++ 所提供的功能。
  - ◆ 代码的经济性 - 使用 C++ 的一些特性，如 RTTI, 模板，和多重继承等等，以使得代码更短也更易于使用。
  - ◆ 各个类定义版本的无关性。即当一个类的定义更改时，旧文件仍可导入到新版本的类中。
  - ◆ 深的指针保存和恢复。即指针的保存与恢复分别保存和恢复所指的数据。
  - ◆ 对共享数据指针的正确恢复。
  - ◆ STL 容器及其它常用模板的序列化。
  - ◆ 数据的可移植性 - 在一个平台上创建的字节流可以在另一平台上读出。

- 序列化：
  - ◆ 关于 serialization （序列化）
  - ◆ 使用 boost.serialization

- 关于 class archive 和 function serialize()、save()、load()
  - ◆ archive 有 output archive 和 input archive ,
  - ◆ output archive 类似于输出数据流, 数据可以通过 & 或 << 操作符保存到 archive 中
  - ◆ input archive 类似于输入数据流, 数据可以通过 & 或 >> 操作符从 archive 中取出
  - ◆ 对于简单数据类型 (基本类型) 可以直接调用 & 或 <<、>> 将数据放入 archive 或从 archive 中取出
  - ◆ 对于自定义类型, 则需通过 serialize() 或 save() 和 load() 来将数据输出到 archive、或从 archive 中获取数据

- 示例清单 (DEMO) :
  - ◆ 简单示例 1
  - ◆ 简单示例 1 的非侵入 (Intrusive) 版本
  - ◆ 可序列化的成员
  - ◆ 派生类的序列化
  - ◆ 指针 (包括智能指针) 的序列化
  - ◆ 数组的序列化
  - ◆ STL 容器 (包括 string) 的序列化
  - ◆ 识别类的版本
  - ◆ load() 和 save()
  - ◆ xml 格式的 archive
  - ◆ binaray 格式的 archive



- 序列化是对象持久化 (Persistence)、或让对象可以通过网络传输的必须步骤， boost.serialization 则提供了一组简单易用、功能强大的接口，成功的实现了上述功能。
- 除了 boost.serialization 外， C++ 序列化还有其它的一些实现：
  - ◆ CommonC++ Libraries :  
<http://www.gnu.org/software/commoncpp/>
  - ◆ s11n :  
<http://www.s11n.net/>