

## <<Effective STL中文版>>

### 图书基本信息

书名：<<Effective STL中文版>>

13位ISBN编号：9787302126959

10位ISBN编号：730212695X

出版时间：2006-1

出版时间：清华大学

作者：请买家自查

页数：208

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Effective STL中文版>>

### 内容概要

一本经典著作，其实际价值在于说明了STL的有效用法，这本书和EC，MEC一样是伟大的，绝对需要一读的。

本书充满了从实践中总结出来的智慧，它清晰、简明、透彻的风格必将使每一位STL程序员受益匪浅！

著名译者潘爱民最新译作，值得信赖！ 你已经熟悉STL了。

你知道怎样创建容器、怎样遍历容器中的内容，知道怎样添加和删除元素，以及如何使用常见的算法，比如find和sort。

但是你并不满意。

你总是感到自己还不能充分地利用STL。

本该很简单的任务却并不简单；本该很直接的操作却要么泄漏资源，要么结果不对；本该更有效的过程却需要更多的时间或内存，超出了你的预期。

是的，你已经知道如何使用STL了，但是你并不能确定自己是否在有效地使用它。

## &lt;&lt;Effective STL中文版&gt;&gt;

## 书籍目录

- 引言第1章 容器第1条：慎重选择容器类型。  
第2条：不要试图编写独立于容器类型的代码。  
第3条：确保容器中的对象拷贝正确而高效。  
第4条：调用empty而不是检查size()是否为0。  
第5条：区间成员函数优先于与之对应的单元素成员函数。  
第6条：当心C++编译器最烦人的分析机制。  
第7条：如果容器中包含了通过new操作创建的指针，切记在容器对象析构前将指针delete掉。  
第8条：切勿创建包含auto\_ptr的容器对象。  
第9条：慎重选择删除元素的方法。  
第10条：了解分配子(allocator)的约定和限制。  
第11条：理解自定义分配子的合理用法。  
第12条：切勿对STL容器的线程安全性有不切实际的依赖。
- 第2章 vector和string第13条：vector和string优先于动态分配的数组。  
第14条：使用reserve来避免不必要的重新分配。  
第15条：注意string实现的多样性。  
第16条：了解如何把vector和string数据传给旧的API。  
第17条：使用“swap技巧”除去多余的容量。  
第18条：避免使用vector。
- 第3章 关联容器 第19条：理解相等(equality)和等价(equivalence)的区别。  
第20条：为包含指针的关联容器指定比较类型。  
第21条：总是让比较函数在等值情况下返回false。  
第22条：切勿直接修改set或multiset中的键。  
第23条：考虑用排序的vector替代关联容器。  
第24条：当效率至关重要时，请在map::operator[]与map::insert之间谨慎做出选择。  
第25条：熟悉非标准的哈希容器。
- 第4章 迭代器.....

## &lt;&lt;Effective STL中文版&gt;&gt;

## 媒体关注与评论

带着审视的目光 回顾STL的使用 ——评Effective STL 评论人：Alan Bellingham from ACCU

等级：高度推荐（Highly Recommended） 针对C++的标准库的一部分——标准模版库

（Standard Template Library, STL）的书已经有好几本了。

它们中的一些已经在以前的ACCU书评中获得了“高度推荐”，并且它们名副其实。

那么这本书是为谁而写的呢？

又为什么要写呢？

让我先来告诉你这本书不属于哪类。

它不是指南，如果你还并不知道如何使用STL，那么从本书中也无法学会。

它也不是一本覆盖STL的边边角角的参考书。

对于这种参考书，我推荐两本：Nicolai Josuttis的The C++ Standard Library 和Angelika Langer Standard C++ IOStreams and Locales。

然而，如果你已经使用过STL一段时间，如果你已经了解容器、迭代器和算法结合到一起方法，并且如果你喜欢当自己知道还有很多东西可以学习时所拥有的那种表现力，那么本书非常适合你。

本书的目标在它的副书名中已经说清楚了：50条提高STL应用能力的具体方法。

就像Soctt Meyers以前的书Effective C++和More Effective C++一样，本书也是以一组短小精炼的文章的方式组织的，尽管这些文章被称为条款（item），但它们每一个都是独立完整的。

它们被组织为7章：容器、vector和string、关联容器、迭代器、迭代器、函数子、函数子类、函数及其他。

每一章有一个大概半页的简短介绍，文章剩下的内容通常有两到三页，而当作者深入说明某些问题的时候，篇幅会多达12页甚至更多。

即便在最简单的文章里，如“条款44 容器的成员函数优先于同名的算法”，作者也很仔细地说明了他为什么要给出这个建议，以及何时不要采用这个建议。

这不是一组盲目遵从的规则，作者情愿你能理解他为什么说到这条规则，以便你能够判断什么时候不要遵从。

这就是一个单纯的讲师和一位真正的老师之间的区别。

实际上，我发现作者对C++标准的态度令人耳目一新，他在尊重的同时指出了STL的设计中的不好的选择。

本书从表现形式上也安排得很好（指原版），值得一提的是，使用了简洁明了的色彩来说明代码中的特别指出。

我们常常会认为所看到的代码的一般形式都是很不明显的。

本书中，这些地方会用红色印刷。

这样显得清晰而不俗气，我觉得其他的书应该效仿。

本书中有一两处印刷错误（指原版书），但是无伤大雅。

而且这些错误并不多见，在正文中并没有出现，那都是经过专家的技术审阅的。

另外，还有几个话题确实没有设计到。

例如，关于STL的书通常不会涉及iostreams，但是，这也许只是意味着还将有一本More Effective STL要出版。

我当然希望是这样，这本书已经让我回过头去审视我使用STL的经历了。

总的来说，如果每天基本上都要用到STL，并且如果你希望用的更好，那么我强烈推荐你现在就去买一本Effective STL。

如果你还不相信我，可以自己去网上查看一些条款。

## 编辑推荐

你已经熟悉STL了。  
你知道怎样创建容器、怎样遍历容器中的内容，知道怎样添加和删除元素，以及如何使用常见的算法，比如find和sort。  
但是你并不满意。  
你总是感到自己还不能充分地利用STL。  
本该很简单的任务却并不简单；本该很直接的操作却要么泄漏资源，要么结果不对；本该更有效的过程却需要更多的时间或内存，超出了你的预期。  
是的，你已经知道如何使用STL了，但是你并不能确定自己是否在有效地使用它。

<<Effective STL中文版>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>