



# Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library

*Scott Meyers*

[Download now](#)

[Read Online](#) 

# Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library

Scott Meyers

**Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library** Scott Meyers  
"This is *Effective C++* volume three - it's really that good."

- Herb Sutter, independent consultant and secretary of the ISO/ANSI C++ standards committee "There are very few books which all C++ programmers *must* have. Add *Effective STL* to that list."

- Thomas Becker, Senior Software Engineer, Zephyr Associates, Inc., and columnist, *C/C++ Users Journal* C++'s Standard Template Library is revolutionary, but learning to use it well has always been a challenge. Until now. In this book, best-selling author Scott Meyers ( *Effective C++* , and *More Effective C++* ) reveals the critical rules of thumb employed by the experts - the things they almost always do or almost always avoid doing - to get the most out of the library. Other books describe *what's in* the STL. *Effective STL* shows you *how to use it*. Each of the book's 50 guidelines is backed by Meyers' legendary analysis and incisive examples, so you'll learn not only what to do, but also *when* to do it - and *why*. Highlights of *Effective STL* include:

Advice on choosing among standard STL containers (like vector and list), nonstandard STL containers (like hash\_set and hash\_map), and non-STL containers (like bitset).

Techniques to maximize the efficiency of the STL and the programs that use it.

Insights into the behavior of iterators, function objects, and allocators, including things you should *not* do.

Guidance for the proper use of algorithms and member functions whose names are the same (e.g., find), but whose actions differ in subtle (but important) ways.

Discussions of potential portability problems, including straightforward ways to avoid them. Like Meyers' previous books, *Effective STL* is filled with proven wisdom that comes only from experience. Its clear, concise, penetrating style makes it an essential resource for every STL programmer.

## Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library Details

Date : Published June 16th 2001 by Addison-Wesley Professional (first published January 1st 2001)

ISBN : 9780201749625

Author : Scott Meyers

Format : Paperback 288 pages

Genre : Computer Science, Programming, Technical, Science, Nonfiction, Computers, Software

 [Download Effective STL: 50 Specific Ways to Improve Your Use of ...pdf](#)

 [Read Online Effective STL: 50 Specific Ways to Improve Your Use o ...pdf](#)

**Download and Read Free Online Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library Scott Meyers**

---

## From Reader Review Effective STL: 50 Specific Ways to Improve Your Use of the Standard Template Library for online ebook

### Graham says

Although this book doesn't have C++ in the title, those in the know will understand its relationship to the aforementioned programming language - it is about the effective use of the Standard Template Library, a set of algorithms and data structures for use in C++ programs.

---

### Hertz He says

In almost every item in the book presents an opinion or a preference of a particular style of writing programs in C++, the better thing about it, is that Meyers followed that opinion with a deep reasoning on why that's the case. A lot of the reasoning brings us to the perspective of the library/compiler writers. And that only strengthens the arguments made in the book.

Plus, I chuckled every time I read an item. The jokes Scott Meyers made in his "Efficient" series of C++ books are hilarious, in a geeky way.

Excellent book, I love it!

---

### Matt says

I appreciate the practical approach of Scott Meyers' *Effective...* books. Although this book is recommended for intermediate to advanced C++ programmers, I would expand that recommendation to any C++ programmer with a computer science background that has done *any* work with the Standard Template Library (STL). I found some of the topics in this book interesting, and suspect the others may have been more interesting if I had read this several years earlier in my C++ programming experience.

---

### Vladimir Atamanov says

Slightly outdated, but I'm reading "Effective Modern C++" next, so it's ok for me.

---

### Simón says

Excellent book!

Although the name states STL, Scott Meyers focuses mainly on the data structures and algorithms part of it: expect plenty of detail on why and when you should use certain types over others, performance implications,

---

idioms...

Its review of the STL algorithms is very important: there is an important amount of functionality already available for C++ developers that replace most usages of raw loops. Knowing when to use these methods and which predicates must go with them is great.

Finally, this book was released before C++11 was available, so at certain points you will realise that lambdas could solve certain functionality, that some helpers are now deprecated, that hashed containers are now available as first-class citizens, and that convenient `emplace_*` methods can be used. Other than that, this book is perfectly up to date!

Must read for C++ developers!

---

## Steve says

The biggest problem with a programming book is how to use it. The content in this book is rich, deep, and helpful. But I have to admit I was dubious about getting this book because I could see no solid case for it improving my programming in the laid back format of the previous "50 Specific Ways..." books. And what I found was consistent with my original impressions.

The problem with teaching people to use the STL better is that they have to actually have experience using the STL. I don't just mean experience in a class, I mean real-world programming experience (although such a world is undoubtedly very removed from reality). Real world experience with the STL where this book will actually help you, I believe, is so far removed from most software development activities that this book will haunt you more than help you.

The "50 Specific Ways..." series is actually really useful in that it highlights problems you are likely to encounter when using the environment covered by the subject in the most common ways. The problem is that people who actually develop code with the STL code (i.e. really incorporate it) are probably a lot fewer than instructors would like to claim. I have relatively wide experience in programming with C++ including stand-alone simple stuff, C++ in support of visual studio, C++ for DLL applications, C++ as driver code (perhaps C would be better?), and C++ as part of a roughly 1000 person collaboration developing something like 100k new lines of code every week. In my experience none of the more casual programmers - even in the 1000 person collaboration - had any clue of the details that Scott Meyers touches on in this book.

So the biggest dilemma is how to use this book. That actually is an easy question because the sections are well-written and the material is thorough. First read it quickly. Then use it as a desk reference while reading, editing, and writing code that is based on the STL. Start by using the sections where Meyers explains which containers are good for what specific purposes. Then, if your particular application is discussed in the book, re-read that section again really quick before writing code. Then, when you encounter compiler complaints, use the book to help you figure out what is wrong. For the most part the cross-referencing of compiler errors is pretty good, and there is a section that helps diagnose the meaning behind compiler complaints that span several 1000 characters.

As an example, Meyers discusses why you should use `&V[0]` to get a pointer to the start of a vector (see item 16). Now, when I was working with the 1000 person collaboration, this type of code showed up all the time. My impression was that this was actually bad practice, and that they should really have been using `V.begin()`. However, this is not good practice, and Meyers explains why. He also explains the reason you need to use `&V[0]`, and why you need to check `V.empty()` before using it.

## Craig says

Was a good read after C++ For Game Programmers 3 Chapters on STL. Unfortunately, some of the things are now deprecated in C++11. In general it was good to see when and how to use STL correctly. I'm hoping Scott Meyers updates all his Effective books for C++11 soon :)

---

## Franck Chauvel says

Scott discloses fifty common C++ errors with the Standard Template Library (STL), including what container shall you use (and when), how to use iterators, the remove-erase idiom, etc.

I think this book is definitely useful especially as I haven't read any other texts about the STL. Yet, I found it less enlightening than More Effective C++. I found the text easy to read despite its technical nature, and I appreciate the effort to make it as entertaining as C++ code can be.

---

## Prakash says

Mostly relevant even today...!

5/5.

---

## Maya says

A surprisingly fun and easy read for such a potentially dry topic. I wish more programming authors had Meyers' flare for suspense, drama, and humor.

---

## Flavio Perri says

excellent as each meyers' book about c++. This is not a reference book about STL, but we can find answers about best practices on using containers, algorithms and functors.

---

## Elliott Bignell says

This is the third of Meyers' must-have works that any competent C++ developer is likely to have within reach of his keyboard. It is perhaps not as ubiquitous as the two "Effective" books, as some developers think that using STL is not such a good idea, especially those restricted to DOS and its pernicious spawn. (To steal a Meyers coinage.) If you have such a developer on your team, buy them a copy of Visual Basic and send them home immediately. STL is an open standard and is understood industry-wide; STL is available in a

---

variety of implementations from a variety of competing sources; STL is an achievement of rare brilliance and will help you to make your code shorter, more efficient and more expressive. Meyers shows you how.

Meyers' books do not conform to any easy categorisation, and this book is no exception. They are not textbooks. They are not introductory works. They are not reference works. Despite all this, they are unique in the C++ world and you will find only a minority of developers who do not own a couple and refer to them often. They are of unique utility and you ought to have copies. I read mine afresh every year or two. What they are is a catalogue of lessons from a man who is deeply involved with the C++ language, who is involved in development of its published standard and who has a wealth of production experience and knows the pitfalls. STL is the STANDARD template library. If you use a mainstream compiler you probably already have an implementation and you ought to have a good reason before using something else.

The STL basically breaks down into containers, iterators, algorithms and functors and their adapters, and the overall layout of the book reflects this ontology. Additional sections are provided for vector and string, and for associative containers. A final section covers "Programming with the STL", meaning practical use of it in day-to-day code design. Particularly useful is the advice on deciphering compiler errors generated during use of the STL, which can be - er - Byzantine. To say the least. "Salad of identifiers" springs to mind, in fact. Another one in the "Programming" section which I have learned to take to heart is to prefer STL algorithms to hand-written loops.

Some of this material has been overtaken with the advent of the Boost library, which Meyers himself praises highly. Boost's function binders and lambda functions have made working with containers and algorithms simpler, more interesting and expressive, for instance. However, I would strongly advise anyone moving into Boost to familiarise themselves with this material first, as the two technologies interoperate and Boost borrows heavily from the STL. It also requires the greater expertise to master, so this is the obvious next-to-last step.

A highly entertaining and eminently useful book on an aspect of C++ that few of us take best advantage of.

---

### **Xavrin says**

Outdated, not fun and containing a lot of advice you'll never use.

---

### **Klaim says**

C++ dev must-have.

---

### **Gareth says**

Another great "Effective" book. Every book this guy puts out is great and I would recommend this one to any c++ programmer. For a technical book this it is an easy read and extremely informative. It's the sort of book you should read once and then refer to when you screw up and need to know why. You should also probably read before job interviews. 99% of C++ coding questions seem to come from one of these books.

