DevX.com
The know-how behind application development
SEARCH FOR

TODAY'S HEADLINES | ARTICLE ARCHIVE | FORUMS | TIP BANK

BROWSE DEVX
Specialized Dev Zones
Research Center
eBook Library
.NET
Java
C++
Web Dev
Architecture
Database
Security
Open Source
Enterprise
Mobile
Special Reports
10-Minute Solutions
DevXtra Blogs
Slideshow

Related Resources
Pointers: Advanced Concepts in C++
JavaScript Named Language of the Year
NASA Releases 10 Rules for Writing Safe Code
Using Preprocessor Directives in C++
Dart Now a Top 20 Programming Language

`std::array`: The Secure, Convenient Option for Fixed-Sized Sequences

Migrate your fixed-sized sequences to std::array, which offers a secure, efficient, and convenient alternative to built-in arrays—sans the overhead of vector.

by Danny Kalev

Google +

Jun 11, 2009

Page 1 of 3

When implementing a sequence of elements that needs to grow and shrink dynamically, `std::vector` is a fine choice. It provides the convenience of a standard container class, including automatic memory management, random access to elements, a uniform interface, and compatibility with standard algorithms.

However, for a sequence whose size is fixed, the performance hit that comes with all that convenience isn't always justifiable. In particular, a vector incurs the following overhead:

- *Nontrivial initialization and destruction.* The constructor and destructor of `std::vector` are relatively costly. This overhead is particularly noticeable when your program creates a large number of vectors.
- *Allocation time.* Vectors allocate storage for their elements at runtime.
- *Space overhead.* A vector object typically occupies 32 bytes.

Built-in arrays carry none of this overhead. They allocate their storage statically and have zero space overhead. So why not just use built-in arrays instead for your fixed-sized sequences? Built-in arrays have their own downside: Their security loopholes make them an insecure choice.

Luckily, the C++ Standard Library offers a container-like template called `std::array`, which offers a secure, efficient, and convenient alternative to built-in arrays—without the overhead of vectors. This 10-Minute Solution shows how to use this powerful alternative for fixed-sized sequences.

The Problem

How can I implement fixed-sized sequences without the security loopholes of built-in arrays or the noticeable overhead of vectors?

The Solution

Use `std:array` as a secure, efficient, and convenient alternative to built-in arrays and vectors for your fixed-sized sequences.

Next Page

Other Articles by This Author

1 2 3 Next Page »

0 Comments (click to add your comment)

Comment and Contribute

Your name/nickname

Your email

WebSite

Subject

(Maximum characters: 1200). You have 1200 characters left.

Submit Your Comment