

# Exception-Safety in Generic Components

## Lessons Learned from Specifying Exception-Safety for the C++ Standard Library

David Abrahams

Dragon Systems

David.Abrahams@dragonsys.com

**Abstract.** This paper represents the knowledge accumulated in response to a real-world need: that the C++ Standard Template Library exhibit useful and well-defined interactions with exceptions, the error-handling mechanism built-in to the core C++ language. It explores the meaning of exception-safety, reveals surprising myths about exceptions and genericity, describes valuable tools for reasoning about program correctness, and outlines an automated testing procedure for verifying exception-safety.

**Keywords:** exception-safety, exceptions, STL, C++

## 1 What Is Exception-Safety?

Informally, exception-safety in a component means that it exhibits reasonable behavior when an exception is thrown during its execution. For most people, the term “reasonable” includes all the usual expectations for error-handling: that resources should not be leaked, and that the program should remain in a well-defined state so that execution can continue. For most components, it also includes the expectation that when an error is encountered, it is reported to the caller.

More formally, we can describe a component as minimally exception-safe if, when exceptions are thrown from within that component, its invariants are intact. Later on we’ll see that at least three different levels of exception-safety can be usefully distinguished. These distinctions can help us to describe and reason about the behavior of large systems.

In a generic component, we usually have an additional expectation of *exception-neutrality*, which means that exceptions thrown by a component’s type parameters should be propagated, unchanged, to the component’s caller.

## 2 Myths and Superstitions

Exception-safety seems straightforward so far: it doesn’t constitute anything more than we’d expect from code using more traditional error-handling techniques. It might be worthwhile, however, to examine the term from a psychological viewpoint. Nobody ever spoke of “error-safety” before C++ had exceptions.