Making C++ Software *Allocator Aware*

Document #:	P2127R0
Date:	2020-08-21
Project:	Programming Language C++
Reply-to:	Pablo Halpern (on behalf of Bloomberg) <phalpern@halpernwightsoftware.com> John Lakos <jlakos@bloomberg.net>

NOTE: This white paper is intended to motivate continued investment in developing and maturing better memory allocators in the C++ Standard. Although its instructional content is targeted at Bloomberg engineers, the wider C++ community can benefit from this illustration of the use of memory allocators in a large industrial codebase, including the related challenges that the authors are working to alleviate both at Bloomberg and in the C++ Standard.

Abstract

Allocator-aware (AA) software — software that allows a client to supply an allocator at object construction — provides the application developer with an effective, lower-cost alternative to writing bespoke types having individually customized memory management.[1] Creating AA software, however, can be considerably more complex than using existing AA software. After introducing the requirements for an AA type compatible with the BDE[2] Development Environment, this paper walks the reader through the steps of transforming a simple struct into an AA class and then explains how to accomplish this task for increasingly complex categories of types, culminating with container class templates.

[1] Motivational background can be found in [**?**]. Information on using AASI can be found in [**?**].

[2] BDE is an initialism that began as Bloomberg Development Environment and is now understood to simply describe a group within Bloomberg.

Contents

1 Introduction

References