"...one of the most highly regarded and expertly designed C++ library projects in the world."
— Herb Sutter and Andrei Alexandrescu, C++ Coding Standards

Chapter 12. Boost.DLL

Renato Forti

Antony Polukhin

Copyright © 2014 Renato Tegon Forti, Antony Polukhin

Copyright © 2015 Antony Polukhin

Copyright © 2016 Antony Polukhin, Klemens Morgenstern

Distributed under the Boost Software License, Version 1.0. (See accompanying file LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)

Table of Contents

Motivation
Getting started
Tutorial
    Plugin basics
    Factory method in plugin
    Searching for a symbol in multiple plugins
    Linking plugin into the executable
    Symbol shadowing problem (Linux)
    Executing callbacks on library unload
    Querying libraries for symbols
    Advanced library reference counting
    Importing a C function from Windows dll
Mangled Import
    Support & Requirements
    Mangled Import Example
    Class Import
    Overloading qualifiers
Missuses
Reference
    Shared Library Reference
    Shared Library Refcountable Reference
Limitations

Exporting weak symbols (Fails on MinGW, Android)
User defined section names (Fails on SunOS + Oracle Solaris Studio Compilers)
Thread safe library loading (Fails on FreeBSD, MacOS, iOS and some other)
Nested Function Definitions
F.A.Q.
Design Rationale
ABI portability across compilers
User's plugin API
Performance and memory allocations
Self loading
Aliases vs Mangling
Dependencies
Revision History
Acknowledgements

Motivation

Adding a specific features to an existing software applications at runtime could be useful in many cases. Such extensions, or plugins, are usually implemented using Dynamic Library Modules (DLL,SO/DSO) loaded at runtime.

This library was designed to simplify plugin development using C++ in a portable cross-platform manner.

Library provides a portable across platforms way to:

- load libraries
- import any native functions and variables
- make alias names for C++ mangled functions and symbols
- query libraries for sections and exported symbols
- self loading and self querying
- getting program and module location by exported symbol

Last revised: August 19, 2017 at 16:58:58 GMT