



C++ ABI Testsuite

Highlights

CodeSourcery's C++ ABI Testsuite helps to:

- Ensure binary compatibility with other C++ compilers, including the GNU C++ compiler.
- Ensure consistency between compilers with other C++-aware tools.
- Avoid accidental changes to binary interfaces over time.

Introduction

The C Application Binary Interface (C ABI) for a particular platform specifies the conventions used by C programs on the platform. For example, the C ABI specifies the sizes and alignments of the basic data types, whether arguments to functions are passed in registers or on the stack, and the format of object files. Two C compilers that implement the same C ABI produce object files that can be freely intermixed to form a complete program.

The C++ Application Binary Interface (C++ ABI) is the analogous specification for the C++ programming language. The C++ ABI specifies how names of overloaded functions are “mangled” (i.e., how the type of the function is embedded in the function's name), how big

objects of a particular class will be, and what data structures are created to support calls to virtual functions.

In general, the C++ ABI for a particular platform depends upon the C ABI. For example, the C++ ABI specifies sizes and alignments for C++ classes by transforming those C++ classes into C structures and then using the rules set out in the C ABI.

Historically, no two compilers have implemented the same C++ ABI. For example, object files created by the GNU C++ Compiler (G++) were incompatible with object files created by popular workstation and embedded system vendors.

This problem extends beyond compilers to other kinds of tools. For example, debuggers, profilers, linkers, and source analysis tools frequently rely on details of the C++ ABI. Because these tools often support only one C++ ABI on a particular platform, these tools work well with only one compiler on that platform. Because the C++ ABIs used by various vendor compilers contained substantial differences, it was very difficult to write tools that worked well on multiple operating systems.

The Itanium[®] C++ ABI

The Itanium[®]¹ C++ ABI was designed to be the C++ ABI for Intel's Itanium[®] archi-

¹Itanium is a registered trademark of the Intel Corporation.

ture. One goal was to provide a C++ ABI that all Itanium[®] compilers would implement, thereby ensuring compiler interoperability on Itanium[®] systems. Another goal was to provide a C++ ABI that provided unparalleled application performance, especially for C++ constructs that have traditionally been considered expensive, such as virtual function calls and the use of virtual base classes.

CodeSourcery contributed substantially to the design of the C++ ABI. CodeSourcery recognized that the ABI, while originally designed for Itanium[®], would apply equally well to other architectures, ranging from embedded systems to mainframes. Therefore, from this point forward, this document uses the term “C++ ABI” instead of “Itanium[®] C++ ABI.”

CodeSourcery has provided implementations of the C++ ABI to several major system vendors. CodeSourcery’s implementation of the C++ ABI in G++ is the de facto standard C++ ABI on GNU/Linux systems on all hardware architectures. In fact, the GNU C++ compiler uses the C++ ABI on all platforms. Compilers that wish to interoperate with G++ must implement the C++ ABI. Edison Design Group hired CodeSourcery to implement the C++ ABI in its industry-leading C++ front end. That front end is used in a wide variety of compilers and tools.

In short, the C++ ABI is emerging as an industry standard.

CodeSourcery’s C++ ABI Testsuite

The C++ ABI is very complex. It contains some features, such as a very aggressive class layout algorithm, that are extremely difficult to implement correctly. It is possible to make mistakes that do not manifest in ways that can be detected by traditional C++ test suites. For example, if the size of a class is incorrect, no C++ conformance test suite will detect the

problem. Therefore, it has been difficult to ensure that a particular compiler conforms to the C++ ABI.

CodeSourcery’s ABI Testsuite solves the testing problem. A compiler that passes CodeSourcery’s testsuite has been tested for conformance on issues such as:

- object layout
- virtual function tables
- name mangling
- runtime library support.

The Testsuite has a proven track record. CodeSourcery has used the Testsuite to validate G++, and found more than ten critical defects. Each of these defects would have made G++ incompatible with other C++ compilers implementing the C++ ABI. (CodeSourcery has fixed these defects in G++.)

The Testsuite contains thousands of individual tests. Tests can be run individually or in groups. CodeSourcery has packaged the Testsuite with its QMTest testing tool, which provides both command-line and graphical interfaces.

If a compiler fails a particular test, the Testsuite provides detailed information about the cause of the failure. For example, when testing object layout, the Testsuite indicates what base class or data member was placed at an incorrect location and also indicates the correct location.

Platform-Specific Testing

Even a C++ compiler that implements the ABI correctly on one platform can implement it incorrectly on another platform. There are differences in the C++ ABI from platform to platform due to architecture-specific constraints. For example, 32-bit platforms use

different rules from 64-bit platforms. Differing alignment requirements for functions require differing pointer-to-member implementations. The C ABI layout rules influence class layout in critical ways.

Even compliant C++ front ends must be configured correctly to precisely match the C++ ABI on the target platform. Failure to configure the C++ front end correctly can result in a C++ compiler which passes traditional C++ test suites, but fails to correctly implement the C++ ABI.

Therefore, CodeSourcery strongly recommends that all compiler and tool vendors use the Test suite on all platforms on which their tools will be deployed.

Licensing

CodeSourcery's C++ ABI Test suite is licensed on a per-platform basis. There are no per-copy royalties on distributed compilers, and no per-seat license fees. CodeSourcery's license permits the licensee's entire organization to use the Test suite on the licensed platforms. All licensees are automatically provided with

maintenance updates as they become available.

CodeSourcery provides discounts for organizations who wish to license the test suite for use on several platforms simultaneously.

Certification

CodeSourcery certifies ABI compliant compilers. Upon request, CodeSourcery will validate a particular compiler with its test suite and permit the vendor to mark the compiler as "CodeSourcery Certified: C++ ABI Compliant".

Modification

CodeSourcery can modify the C++ ABI Test suite to check for non-standard behavior on a particular platform, if, for example, a vendor is intent on providing backward compatibility with a non-standard variant of the C++ ABI. CodeSourcery can also modify its C++ ABI Test suite to test C++ ABIs that are not derived from the Itanium[®] C++ ABI.

CODESOURCERY C++ ABI TESTSUITE SPECIFICATIONS

FEATURES TESTED

Object Layout
Name Mangling

Virtual Tables
Virtual Table Tables

Pointers to Members
RTTI
Dynamic Arrays
Vague Linkage

Library Support

Calling Conventions

Sizes, alignments, and offsets of objects.
External names used to represent functions, variables, virtual function tables, etc.
Virtual functions, virtual base offsets, virtual call offsets.
Virtual tables during object construction and destruction.
Contents of pointers to members.
Data used by exception handling, dynamic cast, etc.
Padding and alignment of dynamically allocated arrays.
Emission of virtual tables, RTTI structures, etc. in correct locations.
Implementation of guard variable routines, allocation routines, finalization routines.
Correct parameter passing order for constructors, destructors.

USABILITY

Graphical User Interface
Command Line Interface
Failure Diagnosis

Browser-based test execution.
Scriptable interface.
Detailed explanation of failures.

SUPPORTED SYSTEMS

Host Platforms
Target Platforms

Windows, GNU/Linux, UNIX-like operating systems.
ARM EABI, IA32 GNU/Linux, IA64 GNU/Linux.
Other platforms available upon request.

SERVICE & SUPPORT

Support
Maintenance
Certification

Included.
Included.
Available for an additional fee.

LICENSE TERMS

Fees

Company Wide License

Per-platform.
No per-seat license fees, no per-copy royalties.
No site restrictions, no project restrictions.

