

A C++ ABI Test Suite

because

ABI bugs are a NIGHTMARE

Why should you test the ABI?

- To ensure release to release compatibility.
- To ensure compatibility with third party libraries.
- To ensure compatibility with tools that expect a specific ABI.

ABI bugs are a nightmare as they can hit you where you least expect and debuggers are often useless against them.

What does the ABI Test Suite do?

It tests a compiler's implementation against the Itanium C++ ABI specification, by having C++ code that exercises various parts of the ABI specification, and compares the layout generated by the compiler-under-test to the 'correct value'.

Supporting x86 ILP32 and LP64

- All Itanium C++ ABI compilers; clang, gcc and others
- Both native and cross compilers
- Supports quirky compiler configurations
- Uses the lit framework
- Most tests were automatically generated
- Clang TOT passes

Sample code

```
ABI test suite.c  X
// RUN: c_compiler -c %s -I "common" -o %t3.o
// RUN: linker -o %t2.self %t1.o %t2.o %t3.o
// RUN: runtool %t2.self | checker "TEST PASSED"
#include "testsuite.h"
.....
#ifdef __cplusplus
struct efgh : virtual abcd {
    int fld;
    virtual void bar(); // _ZN4efgh3barEv
    efgh(); ~efgh();
};
void efgh::bar(){vfunc_called(this, "_ZN4efgh3barEv");}
efgh::~efgh(){note_dtor("efgh", this);}
efgh::efgh(){note_ctor("efgh", this);}
static void Test_efgh()
{
    extern Class_Descriptor cd_efgh;
    void *lvp;
    ABISELECT(double,int) buf[ABISELECT(3,4)];
    init_test(&cd_efgh, buf);
    efgh *dp, &lvp = *(dp=new (buf) efgh());
    ...
    test_class_info(&lvp, &cd_efgh);
}
static Arrange_To_Call_Me vefgh(Test_efgh, "efgh", ABISELECT(16,12));
#else // __cplusplus
extern void _ZN4efghC1Ev();
Name_Map name_map_efgh[] = { NSPAIR(_ZN4efghC1Ev), NSPAIR(_ZN4efghD1Ev), {0,0} };
extern VTBL_ENTRY _ZTI4efgh[];
extern void _ZN4efgh3barEv();
static VTBL_ENTRY vtc_efgh[] = {
    ABISELECT(12,8), 0,
    (VTBL_ENTRY)&_ZTI4efgh[0],
    (VTBL_ENTRY)&_ZN4efgh3barEv,
};
extern VTBL_ENTRY _ZTV4efgh[];
static VTT_ENTRY vtt_efgh[] = { {&_ZTV4efgh[3]}, 3,4, };
Class_Descriptor cd_efgh = { "efgh", // class name
    ..

```

What does it test ?

- Size and alignments of classes
 - Offsets of fields and base classes
 - Bit fields
 - vtbl and VTT contents
 - ctor and dtor vtables
 - Name mangling
 - Empty classes
 - Thunks
 - Init guard variables
 - RTTI /typeinfo vars
- Classes for object layout tests were generated by reading of the spec, exhaustive generation within some parameters, and collecting examples from existing code.
 - Tests were generated by modifying an EDG based compiler to produce C and C++ code.
 - Build the tests with your compiler and run on your target.
 - The test suite consists of slightly over one million unique classes spread over roughly four hundred test files.

Getting started:

Put lit in PYTHONPATH and put FileCheck and clang in PATH
\$ svn co http://llvm.org/svn/llvm-project/test-suite/trunk/ABI-Testsuite
\$ python sample.py test
See *README.txt*, *FAQ.pdf* and the *Design document* for full details.



PlayStation®