

SuperTest

More than compiler validation

from ACE Associated Compiler Experts

Compiler quality and conformance

The complexity of software systems is increasing rapidly, typically doubling the number of source lines by each year. On one hand reuse of existing code, third party libraries and automated code generation ease the software development process, but ultimately this great diversity of source code needs to be compiled into target code. Clearly more important than performance, compiler quality is imperative. And both compiler vendor and end-user need to be certain that their compilers are up to the task, for a single compiler issue can cause immense problems for a multitude of programmers and software projects. In this light, the investments in a stringent compiler quality assurance system will easily outweigh any possible causal damage from a compiler producing wrong code.

Determining the quality and robustness of C/C++ compilers is a complicated and challenging task. This is because compilers are complex by nature, translating source code into highly optimized output code without jeopardizing the reliability. Furthermore C and C++ are used in a wide range of applications, from small embedded systems to large distributed database applications, often leading to most ingenious code constructs. Additionally many third- and fourth generation application generator tools produce code that is very often unlike that which a human programmer would write.

SuperTest

SuperTest is the world's most comprehensive compiler test and validation suite, coming from ACE Associated Compiler Experts in Amsterdam. With SuperTest, ACE makes its 30 years of experience and expertise in compiler construction and testing available in the form of a compiler test and validation suite that goes well beyond the odd language conformance suite. Both professional compiler developers and software quality assurance engineers alike will appreciate the over sixteen thousand source files in SuperTest, providing over 800,000 conformance tests, as well as a host of quality regression tests dealing with compiler internals such as analyses and optimizations. And as compilation techniques evolve, this set is constantly growing. The procedures include both positive and

negative tests and allow for selective testing of test subsets and previously failed tests.

The straightforward user interface seamlessly integrates into the compiler development environment, allows easy addition of new tests and provides cross-validation support.

The language conformance checks included in SuperTest check the compiler against ISO/IEC 9899 C, ISO/IEC 14882 C++, and ISO/IEC TR 18037 Embedded C specifications. The included tests verify the compiler using a comprehensive collection of compiler issues that is grown from decades of compiler construction and validation at ACE, as well as from the community of CoSy users. The SuperTest depth test suite verifies the compiler's ability to correctly handle combinations of function calls, operators, storage classes, data types and pointer indirection levels. The built-in regression support eases tracking of progress towards validation and catches quality lapses.

This collection of expert compiler validation tools makes SuperTest the most versatile compiler test suite for the professional compiler developer. In order to reduce compiler verification efforts and delays in the release schedule, compiler testing should best start right from the first day of development, well before external release and continue throughout

the lifetime of the compiler.

SuperTest is equally suitable for the end-user who needs to understand or validate the quality of the compiler(s) to be used for a critical software project.

```
#define COUNT      6
int clear[COUNT] = {
    2, 3, 5, 7, 11, 13
};

int main(int argc, char * argv[])
{
    int    i;
    int    bits;
    int    c;

    bits = 0xffff;

    for (i = 0; i < COUNT; i++) {
        c = clear[i];
        bits &= ~(1 << c);
    }

    return(bits == 0xd753);
}
```



SuperTest

- C/C++ compiler test and validation suite
- Based upon 30+ years of experience in compiler construction and validation
- Unique test coverage
- Constantly growing
- Regression tests
- Cross-validation
- Reduced compiler validation effort
- Seamless integration in compiler development environment
- Clear reporting

Industry feedback

- "... By selecting just the ANSI C parts of the test suite we were able to rapidly check that our initial compiler was working as we expected. Problems highlighted by SuperTest have usually allowed us to home in on the cause of the problem quite quickly. SuperTest provides confidence in the scope and stability of our compiler. We view it as the prime step into testing our compiler. Regular use of SuperTest identifies regression problems in a timely manner. We use SuperTest at least weekly and more frequently when making significant changes to our compiler."
- "... Through its systematic testing, compiler stressing and corner cases, the SuperTest tests revealed possibly hundreds of issues."
- "... The test cases are well documented, which tell exactly where the problem is."
- "... SuperTest is a really valuable tool for us. The verbose mode is wonderful for pointing me at the failing section of code. SuperTest found us implementation holes, implementation bugs, documentation errors, simulation bugs and many regressions. It is my primary regression test bench. I'd start using it very early in future compiler projects."
- "... The suite detected a lot of errors during our developments. Running the test suite successfully gave us enough confidence about the acceptable quality of each compiler version to be released to a next stage."

Overview

- Over 16 thousand source files
- Over 800,000 conformance and quality checks
- Positive and negative tests
- ISO/IEC 9899 (C)
- ISO/IEC 14882 (C++)
- ISO/IEC TR 18037 (Embedded C)
- DSP-C
- Coverage tests
- Depth test suite
- Regression testing
- Easy to install
- Clean, easy, interactive/batch user interface
- Reports in plain text and HTML format
- Requirements: UNIX/Linux host operating system
- Source site licensing model

Statistics	LOG-00 bit	LOG-01 bit	LOG-02 bit	LOG-03 bit	LOG-04 bit	LOG-04
Compile errors	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed
2.1	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4	0/0/4
2.2	0/0/32	0/0/32	0/0/32	0/0/32	0/0/32	0/0/32
3.0	0/0/253	0/0/253	0/0/253	0/0/253	0/0/253	0/0/253
3.1	0/3/119	0/3/119	0/3/119	0/3/119	0/3/119	0/3/119
3.2	0/0/342	2/5/447	0/0/342	0/0/342	2/6/446	2/6/446
3.3	0/1/6	0/1/6	0/1/6	0/1/6	0/1/6	0/1/6
3.4	1/7/226	1/7/226	1/7/226	1/7/226	1/7/226	1/7/226
3.5	0/0/175	0/0/175	0/0/175	0/0/175	0/0/175	0/0/175
3.6	0/0/20	0/0/20	0/0/20	0/0/20	0/0/20	0/0/20
3.7	0/1/92	0/1/92	0/1/92	0/1/92	0/1/92	0/1/92
3.8	0/0/218	0/0/218	0/0/218	0/0/218	0/0/218	0/0/218
4	0/0/2668	0/0/2668	0/0/2668	0/0/2668	0/0/2668	0/0/2668
major	3/26/4602	3/26/4602	3/26/4602	3/27/4601	4/27/4600	4/27/4600
Subtotal	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed	failed/check/passed
Run errors	0/0/3	0/0/3	0/0/3	0/0/3	0/0/3	0/0/3
2.1	0/0/31	0/0/31	0/0/31	0/0/31	0/0/31	0/0/31
2.2	0/0/253	0/0/253	0/0/253	0/0/253	0/0/253	0/0/253
3.0	0/0/74	0/0/74	0/0/74	0/0/74	0/0/74	0/0/74
3.1	0/0/381	1/0/381	0/0/381	1/0/381	1/0/381	1/0/381
3.2	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5
3.3	1/0/146	1/0/146	1/0/146	1/0/146	1/0/146	1/0/146
3.4	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5	0/0/5
3.5	1/0/162	1/0/162	1/0/162	1/0/162	1/0/162	1/0/162
3.6	0/0/11	0/0/11	0/0/11	0/0/11	0/0/11	0/0/11
3.7	0/0/73	0/0/73	0/0/73	0/0/73	0/0/73	0/0/73
3.8	9/0/209	9/0/209	9/0/209	9/0/209	9/0/209	9/0/209
4	10/0/2658	10/0/2658	10/0/2658	15/0/4353	15/0/4352	10/0/4367
major	21/0/4347	21/0/4347	18/26/4587	15/0/4353	19/27/4585	10/27/4594
Subtotal	24/26/4581	24/26/4581	18/26/4587	18/27/4586	19/27/4585	10/27/4594
Total	LOG-00 bit	LOG-01 bit	LOG-02 bit	LOG-03 bit	LOG-04 bit	LOG-04
Regression	failed	failed	failed	failed	failed	passed
3.3.2.2.tellipen	passed	passed	passed	passed	passed	passed
3.3.2.2.xpm5020a	failed	failed	failed	failed	failed	failed
3.5.4.truck	passed	passed	passed	passed	passed	passed
3.6.5.3.tape4981	failed	failed	failed	failed	failed	passed
4.0.tape1182	failed	failed	failed	failed	failed	passed
4.0.tape2003	failed	failed	failed	failed	failed	passed
4.0.6.5.12	failed	failed	failed	failed	failed	passed
4.0.6.5.tape4261	failed	failed	failed	failed	failed	passed
4.0.6.5.tape4265	failed	failed	failed	failed	failed	passed
4.0.6.5.tape4349	failed	failed	failed	failed	failed	passed
4.0.6.5.tape4350	failed	failed	failed	failed	failed	passed