The Suite by TFJ
A Validation Suite for the Java™ Language and Core Libraries

Overview

The Suite is a validation suite for the Java™ programming language and core libraries developed in a clean room environment

The Suite is a set of over 8,500 Java programs, comprising over two million lines of code and over 800,000 executable items for testing and evaluating both Java language compilers and Java Virtual Machines (JVMs).

Where applicable the executable classes supplied as part of **The Suite** can be run either as stand-alone applications or as applets. This allows the testing of virtual machine capability in a number of scenarios including the ubiquitous web browser. Test cases can be run as standalone tests or grouped together into collections.

The Suite has three main uses:

- Language conformance testing
- Class Library conformance testing
- Virtual Machine robustness testing

Conformance testing:
Tests that have been written to measure a compiler conformance with the appropriate specification for the language and core library classes. The specifications used are detailed below.

Java Language Conformance

The Java™ Language Specification (Gosling, Joy, and Steele. Addison-Wesley, ISBN 0-201-63451-1, August 1996) and Java™ Language Specification Second Edition ISBN 9-201-31008-2)

The library packages covered are java.lang, java.lang.reflect, java.util, java.util.zip and java.io

Java Class Library Conformance

Tests have been written for the original core library as described by the JLS, with all the packages upgraded to at least the java 1.1 level. Where appropriate we have configurable tests to give either Java 1.1 or Java 2 results.

The library packages tested are java.lang, java.lang.reflect, java.util, java.util.zip and java.io.

Sources for specification for these Class libraries have been Java Doc HTML for the Java 1.3.0 Standard Edition, the Java Class Libraries Second Edition ISBN 0-201-31002-3, 0-201-48552-4, 0-201-31003-1, and the original Java Language Specification.

VM Robustness

In addition to the above tests there are two additional sets of programs specifically aimed at testing VMs.

EXPRESSO
Expresso is composed of a series of Java programs of arbitrary complexity. The aim of these programs is to test the expression generation capability of a Java compiler. Each complex expression has its value calculated from the simpler components that make it up.

For example, a compiler generating code for the expression (a*b) + (c*d) might have an error in keeping track of values and get the wrong answer. But calculated as:
temp1 = a*b;
temp2 = c*d;
temp1 + temp2;
The correct answer is more likely.

By decomposing a complex expression into simpler pieces, EXPRESSO expects to get the right answer and uses that to check the compiler's result on the full complex expression. In addition comments show the breakdown of the calculation.

GRINDER
A large number of self-checking Java programs that test permutations of operators, primitive and reference data types.
Test Case Nomenclature

Types of Test Cases

Positive test cases - identified by suffix _x
These are tests that should compile and run successfully. They test valid assertions of the specifications.

Negative tests - identified by the suffix _z
These are tests that check a compiler's capability to detect bad code. A compiler error(s) should be generated during compilation of the test case. Negative tests only exist as Language conformance test cases

Constructor tests - identified by the prefix CX_
These are short hand names of the constructors for the class under test.

Other suffixes
Indicates a support class that is non-executable

Language Conformance test cases

Each test is derived from a specific statement in the Java Language Specification. To identify the origin of the specification of a test case, each test is identified using a Chapter-Section-Paragraph reference. The current references refer to The Java™ Language Specification (Gosling, Joy, and Steele. Addison-Wesley, ISBN 0-201-63451-1, August 1996)

For example,
j38p9_x denotes Chapter 3, Section 3.8, paragraph 9; and
j161_12p31_x denotes Chapter 16, Section 16.1, paragraph 12, sentence 31

If a paragraph gives rise to multiple test cases, then these are identified by a lower case letter appended to the designation,
e.g. j161_14p13b_x and j161_14p13c_x

Library Conformance test cases

Tests for the original core library as described by the JLS, with the packages upgraded to at least the java 1.1 level. Where appropriate we have configured tests to give either Java 1.1 or Java 2 results.

The naming convention used is simple with the name of each test case reflecting the name of the method under test.

For example if you wish to find test cases relating to the method read in class BufferedInputStream in package io
simply go to the directory: \lib\io\BufferedInputStream

There may appear to be several test cases for a specific method. This happens when the method under test is overloaded. and the test case nomenclature is numerically sequenced for brevity. In the example for read

Test case name	Method
read_01_x	read()
read_02_x	read(byte[] b)

Example Test Case Output and Reporting
An example of output from a test program

```
*** TFJ: Test For Java(tm), Util(701,JDKLevel=130)
#Begin Case (jls.ch3.j31p1_x)
#Reached first test (line 49)

#End Case: jls.ch3.j31p1_x,3,0

*** 3 Successful test items in j31p1_x ***
*** 0 Errors detected in j31p1_x ***
*** 3 Total test items in j31p1_x ***
```

However, if there is an error during execution of the program then the output will have additional information:

```
*** TFJ: Test For Java(tm), Util(701,JDKLevel=130)
#Begin Case (jls.ch15.j15_251p71_x)
#Reached first test (line 113)
ERROR in j15_251p71_x at line  114 : (1a2b) != (1a2b3b)
ERROR in j15_251p71_x at line  134 : (1b2b) != (1b2b3b)

#End Case: jls.ch15.j15_251p71_x,4,2

*** 4 Successful test items in j15_251p71_x ***
*** 2 Errors detected in j15_251p71_x ***
*** 6 Total test items in j15_251p71_x ***
```

It is possible to review the source code to find the reason for the error. Further diagnostic traces can be enabled to allow particular locations of problems to be identified.

All the executable test items can be run on a Java enabled browser, security manager permitting.
Results Presentation
Final result tabulation, summary and reporting are presented in both HTML and comma delimited text files. Hyper links within the HTML output allow easy visual review of test output and source code.
Typical Users

Java Developers

All Java compilers are not the same. There is evidence of Java compilers differing dramatically in both their error diagnostic capability and the correctness of their code generation. The only way to make an informed decision on which compiler is the right one for you is from the output of an industrial strength test suite from an independent organisation such as TFJ.

Compiler Quality Assurance

The Suite is a test tool for checking conformance against Java Language Specification and for overall robustness and correctness. **The Suite** provides a collection of test items with well over 8,500 Java programs. **The Suite** can supplement internally developed test suites giving you a true clean room approach to testing. The output from **The Suite** can be incorporated into existing reports either via HTML or via a spreadsheet.

Java Virtual Machine

The Suite provides several thousand self-checking executable class files for stressing a VM. During **The Suite** development, we have identified variations in execution of class files between VMs from the same supplier on different platforms as well as numerous traditional bugs.

Licencing

TFJ has no business association with any compiler maker or vendor, and provides informed, but unbiased, evaluation tools.

TFJ offers simple and convenient **Single** Site Source Code License Agreements. There is no limit on the number or type of machines on which the licensed materials may be located, simply that the machines must be owned by the licensee and be located within a Single Site. Our licence agreement explains this in detail.

Distributor

Please contact The Test Place, for product, licensing, and pricing information.

The Test Place Limited
PO Box 506
Milton Keynes
Bucks
United Kingdom
MK7 8JD
info@thetestplace.co.uk
TEL +44 (0)1908-645-717
FAX +44 (0)870-054-8815

http://www.thetestplace.com