JVS® - The Plum Hall Validation Suite for Java™ (including CLDC)

Brief Description

JVS® is the Plum Hall validation suite for the Java™ programming language. **JVS** is a set of over 9000 Java programs, comprising over 2.7 million lines of code and over 800,000 executable items for testing and evaluating both Java language compilers and Java Virtual Machines (JVMs). The JVS has three main sections:

- Language conformance tests
- Class library conformance test
- Stress tests

JVS gives you the test capability for more than just a Java compiler, it comes with class files ready for you to test your virtual machine.

Applet and Application

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.

Regarding JVS01a

The JVS01a release, as of September 2001, provides our corrections for all of the bugs that were reported as of that date. It includes configuration options for those situations in which the observed 1.3 behavior differs from 1.1

Testing CLDC

The JVS01a release includes tools to test your CLDC implementation against any level of CLDC, from the various 1.1 versions through 1.2 and 1.3 or later.

Test Case Notation

JVS has two primary forms of test case notation, when targeting language conformance each test is derived from a specific statement in the Java Language Specification; each statement is identified using Section-Paragraph-Sentence (SPS) notation. For example, j511p17_x denotes section 5.1.1, paragraph 1, and sentence 7. For example:

Test Case	Section	Paragraph	Sentence	Status
j511p17_x	5.1.1	1	7	Executable
j514p14a	5.1.4	1	4	Support class
j5p33_z	5	3	3	Negative test

The alpha letter at the end of a test name indicates the status of both the source and any class generated from the compilation of the Java source file.

> **_x** Indicates that the source is a positive test case and when compiled creates an executable program which should execute indicating a successful outcome.
>
> **a-n** Indicates a support class that is non-executable
>
> **_z** Indicates the source is negative test case and that a compiler error(s) should be generated during compilation of the test case. Whether a resulting class is generated that is executable is unspecified but normally the successful generation indicates that a required diagnostic has not been generated.

Many sentences have multiple test cases, distinguished by a lower case letter appended to the SPS designation, e.g. j832p12a_x and j832p12b_x.

The second form of test case notation is reserved for the class library; it is quite simple. . For every class in the class library that JVS tests, you will find a sub-directory of test cases. For instance, if you are looking to test the class java.lang.Object, then you need to look in the package jvs.lang.Object. There you will find at least one test case for every method in the class. The name of the test case will reflect the name of the method under test.

jvs.lang.Object
CX_default_01_x
GetClass_01_x
GetClass_02_x
ToString_01_x
equals_01_x
HashCode_01_x
clone_01_x
wait_01_x
wait_02_x
wait_03_x
wait_04_x
notify_01_x
notify_02_x
NotifyAll_01_x
NotifyAll_02_x
finalize_01_x

Plum Hall JVS is made up of the following components:

JVS-Grinder

A selection of self-checking Java programs that test permutations of operators, and primitive and data types, the focus of this section is to expose flaws in the arithmetic operator handling of integer and floating point data types.

JVS-Expresso

A selection of Java programs of arbitrary complexity designed to test the expression generation capability of a Java compiler. Expressions are generated using each of the applicable operators, compiled and executed, and calculated results are compared against similarly derived results using the smallest component subexpressions.

Language Validation Tests

JVS contains Language Conformance tests for measuring a compiler conformance against *The Java Language Specification* (JLS) similar to the *conform* and *negtests* of other Plum Hall validation suites. Statements in the JLS have been translated into two categories of tests:

- *positive tests* for valid assertions of statements, which should compile and run successfully, and

- *negative tests* for generation of diagnostics for invalid conditions, checking the compiler's capability to detect bad code.

An example of output from a test program for Chapter 8, section 1, sentence 3 of the JLS looks as follows:

```
*** JVS(R): The Plum Hall Test Suite for The Java(tm) Language, Util(505)
#Begin Case (j813p11_x)
#Reached first test (line 48)

#End Case: j813p11_x

*** 2 Successful test items in j813p11_x ***
*** 0 Errors detected in j813p11_x ***
*** 2 Total test items in j813p11_x ***
```

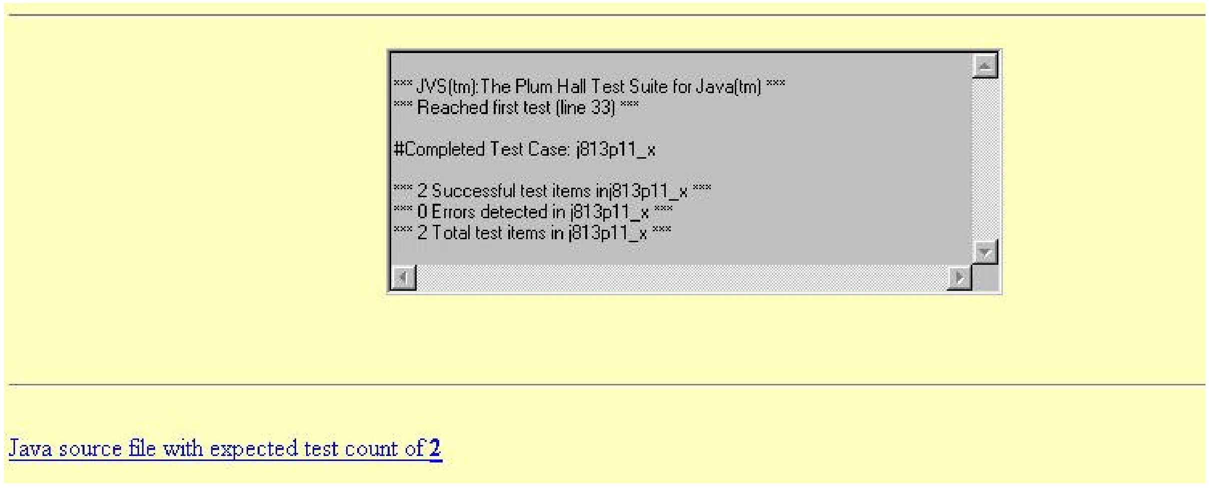
However, if there is an error during execution of the program then the output will have additional information:

```
*** JVS(R): The Plum Hall Test Suite for The Java(tm) Language, Util(505)
*** Reached first test (line 52) ***
ERROR in j55p31_x at line  72 : (-1) != (2147483647)

#Completed Test Case: j55p31_x

*** 22 Successful test items in j55p31_x ***
*** 1 Errors detected in j55p31_x ***
*** 23 Total test items in j55p31_x ***
```

The source code can be reviewed to understand the reason a particular test is giving an error. If required further diagnostic traces can be enabled to allow particular locations of problems to be identified.

All the executable test items can be run by a Java-enabled browser, if permitted by the browser's security manager.

Plum Hall clients desire to test clients' implementations by a black-box comparison against the behavior of various other compilers, including 1.3, 1.2, and 1.1 levels of behavior. In order to accommodate these general requirements, JVS provides a configurable harness named **jruncases**. Also, clients desire to test implementations against the CLDC subset; for this purpose, JVS provides a similarly configurable harness named **truncases.**



```
*** JVS(tm):The Plum Hall Test Suite for Java(tm) ***
*** Reached first test (line 33) ***

#Completed Test Case: j813p11_x

*** 2 Successful test items inj813p11_x ***
*** 0 Errors detected in j813p11_x ***
*** 2 Total test items in j813p11_x ***
```

Java source file with expected test count of **2**

Class Library Validation Tests

The JVS currently has over 3,000 test programs for the original core library as described by the JLS, with the following packages upgraded to 1.1 level (with configuration settings to adapt to 1.2 and 1.3 level).

- java.lang
- java.lang.reflect
- java.util
- java.util.zip
- java.io

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.

Chapter Tests	Test Items	Test Cases
ch3	434	37
ch3neg	150	150
ch4	1590	73
ch4neg	31	31
ch5	812	57
ch5neg	148	148
ch6	601	74
ch6neg	158	158
ch7	127	33
ch7neg	31	31
ch8	629	121
ch8neg	236	236
ch9	233	30
ch9neg	73	73
ch10	262	36
ch10neg	21	21
ch11	204	32
ch11neg	57	57
ch12	44	28
ch12neg	1	1
ch13	825	280
ch13neg	1	1
ch14	437	221
ch14neg	59	59
ch15	3975	433
ch15neg	372	372
ch16	423	245
ch16neg	225	225
ch17	206	31
ch17neg	1	1
chd	357	64
chdneg	48	48
chjb	7	7
chjbneg	14	14
grinder	4936	60
Totals	**17728**	**3488**

Expresso Tests	Test Items	Test Cases
arrar	99036	304
array1	171612	532
array2	171612	532
func	109476	426
method	51156	198
phclass	171612	532
Total	**774504**	**2524**

Library Tests	Test Items	Test Cases
java.lang	3360	1073
java.lang.reflect	631	166
java.util	2693	341
java.util.zip	4447	178
java.io	25697	1413
Total	**36828**	**3171**

Jvs01a	Test Items	Test Cases
Chapter Tests	**17728**	**3488**
Expresso Tests	**774504**	**2524**
Library Tests	**36828**	**3171**
Total	**829060**	**9183**

Figure 1 Example Test Suite Summary of Results

JVS addresses all the language-specification chapters of the Java Language Specification, as well as various HTML documents that described 1.1 changes, plus the various changes made to subsequent JDK implementations. Plum Hall is independent of any compiler maker or vendor, and provides an informed, but unbiased, quality evaluation tool. Plum Hall offers the simple and convenient Plum Hall JVS Single Site Source Code License Agreement—restricted to a Single Site's "two-mile-radius," with no limit on the number or types of destination machines within the radius.

Contact Plum Hall, for product, licensing, and pricing information.

Java Developers

If you are responsible for the selection of a Java compiler for the development of your next generation of products don't be fooled, all Java compilers are not the same. Already there is evidence of Java compilers differing dramatically in both their error diagnostic capability and the correctness of their code generation. The only way you can 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 Plum Hall.

Compiler Quality Assurance

If you are responsible for the quality assurance of the next generation of Java compilers then you will need a test tool for checking conformance against Java Language Specification and for overall robustness and correctness. **JVS** provides a collection of test items—currently over 2,700,000 lines of Java source code, over 800,000 executable test items and over 9000 Java programs. JVS can supplement internally developed test suites with a fresh perspective from outside the internal cultural traditions of your organisation giving you a true clean room approach to testing. The output from JVS can be incorporated into existing reports either via HTML or via a spreadsheet.

Java Virtual Machine

If you are responsible for the quality of a Java Virtual Machine then JVS will give you several thousand self-checking executable class files for stressing your JVM. Already, during our suite development, we have identified variations in execution of class files between JVMs from the same supplier on different platforms. In addition we have discovered numerous traditional bugs.

Plum Hall, Inc.,
3 Waihona Box 44610
Kamuela Hawaii 96743 USA
sales@plumhall.com TEL +1-808-882-1255 FAX +1-808-882-1556
http://www.plumhall.com