Suite#™ - The Plum Hall Validation Suite for C#

**Brief Description**

Suite#™ is the Plum Hall validation suite for the C programming language. Suite# is a set of over 4000 C# programs, for testing and evaluating both C# language compilers and the generated Intermediate Language (IL).

**Test Case Notation**

Suite# has two primary forms of test case notation, when targeting language conformance each test is derived from a specific statement in the C# Language Specification; each statement is identified using Section-Paragraph-Sentence (SPS) notation. For example, z911p17_x denotes section 9.1.1, paragraph 1, and sentence 7. For example:

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Section</th>
<th>Paragraph</th>
<th>Sentence</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>z911p17</td>
<td>9.1.1</td>
<td>1</td>
<td>7</td>
<td>Executable</td>
</tr>
<tr>
<td>z9p33_z</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>Negative test</td>
</tr>
</tbody>
</table>

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 C# source file.

- **No suffix** Indicates that the source is a positive test case and when compiled creates an executable program which should execute indicating a successful outcome.

- **_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. z83p12a and z83p12b.

**Language Validation Tests**

Suite# contains Language Conformance tests for measuring a compiler conformance against ECMA Standard 334 *The C# Language Specification (CLS)* similar to the *conform* and *negtests* of other Plum Hall validation suites. Statements in the CLS 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 2, paragraph 6, sentence 1 of the CLS looks as follows:

*** ZVS(R): The Plum Hall Test Suite for The C#(tm) Language, Util(505) ***
#Begin Case (z82h61)
#Reached first test (line 48)

#End Case: z82h61

*** 2 Successful test items in z82h61 ***
*** 0 Errors detected in z82h61 ***
*** 2 Total test items in z82h61 ***

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

*** ZVS(R): The Plum Hall Test Suite for The C#(tm) Language, Util(505) ***
*** Reached first test (line 52) ***
ERROR in z_1131m32 at line 72 : (3) != (2)

#Completed Test Case: z_1131m32

*** 22 Successful test items in z_1131m32 ***
*** 1 Errors detected in z_1131m32 ***
*** 23 Total test items in z_1131m32 ***

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.

Suite# currently addresses all the language-specification chapters of the C# Language Specification. 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 Suite# 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.

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