

DefaultToStringStyleTest.java - StandardToStringStyleTest.java

```

1/*
2 * Licensed to the Apache Software Foundation (ASF) under one or more
3 * contributor license agreements. See the NOTICE file distributed with
4 * this work for additional information regarding copyright ownership.
5 * The ASF licenses this file to You under the Apache License, Version 2.0
6 * (the "License"); you may not use this file except in compliance with
7 * the License. You may obtain a copy of the License at
8 *
9 * http://www.apache.org/licenses/LICENSE-2.0
10 *
11 * Unless required by applicable law or agreed to in writing, software
12 * distributed under the License is distributed on an "AS IS" BASIS,
13 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
14 * See the License for the specific language governing permissions and
15 * limitations under the License.

```

```

16 */
17package org.apache.commons.lang.builder;
18

```

```

1/* =====
2 * The Apache Software License, Version 1.1
3 *
4 * Copyright (c) 2002 The Apache Software Foundation. All rights
5 * reserved.
6 *
7 * Redistribution and use in source and binary forms, with or without
8 * modification, are permitted provided that the following conditions
9 * are met:
10 *
11 * 1. Redistributions of source code must retain the above copyright
12 * notice, this list of conditions and the following disclaimer.
13 *
14 * 2. Redistributions in binary form must reproduce the above copyright
15 * notice, this list of conditions and the following disclaimer in
16 * the documentation and/or other materials provided with the
17 * distribution.
18 *
19 * 3. The end-user documentation included with the redistribution, if
20 * any, must include the following acknowledgement:
21 * "This product includes software developed by the
22 * Apache Software Foundation (http://www.apache.org/)."
23 * Alternately, this acknowledgement may appear in the software itself,
24 * if and wherever such third-party acknowledgements normally appear.
25 *
26 * 4. The names "The Jakarta Project", "Commons", and "Apache Software
27 * Foundation" must not be used to endorse or promote products derived
28 * from this software without prior written permission. For written
29 * permission, please contact apache@apache.org.
30 *
31 * 5. Products derived from this software may not be called "Apache"
32 * nor may "Apache" appear in their names without prior written
33 * permission of the Apache Software Foundation.
34 *
35 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
36 * WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
37 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
38 * DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
39 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
40 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
41 * LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
42 * USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
43 * ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
44 * OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
45 * OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
46 * SUCH DAMAGE.
47 * =====
48 *
49 * This software consists of voluntary contributions made by many
50 * individuals on behalf of the Apache Software Foundation. For more
51 * information on the Apache Software Foundation, please see
52 * <http://www.apache.org/>.
53 */
54package org.apache.commons.lang.builder;
55
56package org.apache.commons.lang.builder;

```

DefaultToStringStyleTest.java - StandardToStringStyleTest.java

```

19import java.util.ArrayList;
20import java.util.HashMap;
21
22import junit.framework.Test;
23import junit.framework.TestCase;
24import junit.framework.TestSuite;
25import junit.textui.TestRunner;
26
27/**
28 * Unit tests {@link org.apache.commons.lang.builder.DefaultToStringStyleTest}.
29 *
30 * @author <a href="mailto:scolebourne@joda.org">Stephen Colebourne</a>
31 * @version $Id: DefaultToStringStyleTest.java 437554 2006-08-28 06:21:41Z bayard $
32 */
33public class DefaultToStringStyleTest extends TestCase {
34
35    private final Integer base = new Integer(5);
36    private final String baseStr = base.getClass().getName() + "@" +
Integer.toHexString(System.identityHashCode(base));
37
38    public DefaultToStringStyleTest(String name) {
39
40        super(name);
41    }
42
43    public static void main(String[] args) {
44        TestRunner.run(suite());
45    }
46
47    public static Test suite() {
48        TestSuite suite = new TestSuite(DefaultToStringStyleTest.class);
49        suite.setName("DefaultToStringStyle Tests");
50        return suite;
51    }
52
53    protected void setUp() throws Exception {
54        super.setUp();
55        ToStringBuilder.setDefaultStyle(ToStringStyle.DEFAULT_STYLE);
56    }
57
58    protected void tearDown() throws Exception {

```

```

57import java.util.ArrayList;
58import java.util.HashMap;
59
60import junit.framework.Test;
61import junit.framework.TestCase;
62import junit.framework.TestSuite;
63import junit.textui.TestRunner;
64
65/**
66 * Unit tests {@link org.apache.commons.lang.ToStringStyleTest}.
67 *
68 * @author <a href="mailto:scolebourne@joda.org">Stephen Colebourne</a>
69 * @version $Id: StandardToStringStyleTest.java,v 1.1 2002/09/17 22:07:50
scolebourne Exp $
70 */
71public class StandardToStringStyleTest extends TestCase {
72
73    private final Integer base = new Integer(5);
74    private final String baseStr = "Integer";
75
76    private static final StandardToStringStyle STYLE = new StandardToStringStyle();
77
78    static {
79        STYLE.setShortClassName(true);
80        STYLE.setUseIdentityHashCode(false);
81        STYLE.setArrayStart("[");
82        STYLE.setArraySeparator(", ");
83        STYLE.setArrayEnd("]");
84        STYLE.setNullText("%NULL%");
85        STYLE.setSizeStartText("%SIZE=");
86        STYLE.setSizeEndText("%");
87        STYLE.setSummaryObjectStartText("%");
88        STYLE.setSummaryObjectEndText("%");
89    }
90
91    public StandardToStringStyleTest(String name) {
92
93        super(name);
94    }
95
96    public static void main(String[] args) {
97        TestRunner.run(suite());
98    }
99
100    public static Test suite() {
101        TestSuite suite = new TestSuite(StandardToStringStyleTest.class);
102        suite.setName("StandardToStringStyle Tests");
103        return suite;
104    }
105
106    protected void setUp() throws Exception {
107        super.setUp();
108        ToStringBuilder.setDefaultStyle(STYLE);
109    }
110
111    protected void tearDown() throws Exception {

```

DefaultToStringStyleTest.java - StandardToStringStyleTest.java

```

58     super.tearDown();
59     ToStringBuilder.setDefaultStyle(ToStringStyle.DEFAULT_STYLE);
60 }
61
62 //-----
63
64 public void testBlank() {
65     assertEquals(baseStr + "[]", new ToStringBuilder(base).toString());
66 }
67
68 public void testAppendSuper() {
69     assertEquals(baseStr + "[]", new
70 ToStringBuilder(base).appendSuper("Integer@8888[]").toString());
71     assertEquals(baseStr + "<null>", new
72 ToStringBuilder(base).appendSuper("Integer@8888<null>").toString());
73     assertEquals(baseStr + "[a=hello]", new
74 ToStringBuilder(base).appendSuper("Integer@8888[]").append("a",
75 "hello").toString());
76     assertEquals(baseStr + "<null>, a=hello", new
77 ToStringBuilder(base).appendSuper("Integer@8888<null>").append("a",
78 "hello").toString());
79     assertEquals(baseStr + "[a=hello]", new
80 ToStringBuilder(base).appendSuper(null).append("a", "hello").toString());
81 }
82
83 public void testObject() {
84     Integer i3 = new Integer(3);
85     Integer i4 = new Integer(4);
86     assertEquals(baseStr + "<null>", new
87 ToStringBuilder(base).append((Object) null).toString());
88     assertEquals(baseStr + "[3]", new
89 ToStringBuilder(base).append(i3).toString());
90     assertEquals(baseStr + "[a=<null>]", new ToStringBuilder(base).append("a",
91 (Object) null).toString());
92     assertEquals(baseStr + "[a=3]", new ToStringBuilder(base).append("a",
93 i3).toString());
94     assertEquals(baseStr + "[a=3, b=4]", new ToStringBuilder(base).append("a",
95 i3).append("b", i4).toString());
96     assertEquals(baseStr + "[a=<Integer>]", new
97 ToStringBuilder(base).append("a", i3, false).toString());
98     assertEquals(baseStr + "[a=<size=0>]", new
99 ToStringBuilder(base).append("a", new ArrayList(), false).toString());
100    assertEquals(baseStr + "[a=[]]", new ToStringBuilder(base).append("a", new
101 ArrayList(), true).toString());
102    assertEquals(baseStr + "[a=<size=0>]", new
103 ToStringBuilder(base).append("a", new HashMap(), false).toString());
104    assertEquals(baseStr + "[a={}]", new ToStringBuilder(base).append("a", new
105 HashMap(), true).toString());
106    assertEquals(baseStr + "[a=<size=0>]", new
107 ToStringBuilder(base).append("a", (Object) new String[0], false).toString());
108    assertEquals(baseStr + "[a=[]]", new ToStringBuilder(base).append("a",
109 (Object) new String[0], true).toString());
110 }
111
112 public void testLong() {

```

```

110     super.tearDown();
111     ToStringBuilder.setDefaultStyle(STYLE);
112 }
113
114 //-----
115
116 public void testBlank() {
117     assertEquals(baseStr + "[]", new ToStringBuilder(base).toString());
118 }
119
120
121 public void testObject() {
122     Integer i3 = new Integer(3);
123     Integer i4 = new Integer(4);
124     assertEquals(baseStr + "[%NULL%]", new
125 ToStringBuilder(base).append((Object) null).toString());
126     assertEquals(baseStr + "[3]", new
127 ToStringBuilder(base).append(i3).toString());
128     assertEquals(baseStr + "[a=%NULL%]", new ToStringBuilder(base).append("a",
129 (Object) null).toString());
130     assertEquals(baseStr + "[a=3]", new ToStringBuilder(base).append("a",
131 i3).toString());
132     assertEquals(baseStr + "[a=3, b=4]", new ToStringBuilder(base).append("a",
133 i3).append("b", i4).toString());
134     assertEquals(baseStr + "[a=%Integer%]", new
135 ToStringBuilder(base).append("a", i3, false).toString());
136     assertEquals(baseStr + "[a=%SIZE=0%]", new
137 ToStringBuilder(base).append("a", new ArrayList(), false).toString());
138     assertEquals(baseStr + "[a=[]]", new ToStringBuilder(base).append("a", new
139 ArrayList(), true).toString());
140     assertEquals(baseStr + "[a=%SIZE=0%]", new
141 ToStringBuilder(base).append("a", new HashMap(), false).toString());
142     assertEquals(baseStr + "[a={}]", new ToStringBuilder(base).append("a", new
143 HashMap(), true).toString());
144     assertEquals(baseStr + "[a=%SIZE=0%]", new
145 ToStringBuilder(base).append("a", (Object) new String[0], false).toString());
146     assertEquals(baseStr + "[a=[]]", new ToStringBuilder(base).append("a",
147 (Object) new String[0], true).toString());
148 }
149
150 public void testLong() {

```

DefaultToStringStyleTest.java - StandardToStringStyleTest.java

```

95     assertEquals(baseStr + "[3]", new
ToStringBuilder(base).append(3L).toString());
96     assertEquals(baseStr + "[a=3]", new ToStringBuilder(base).append("a",
3L).toString());
97     assertEquals(baseStr + "[a=3,b=4]", new ToStringBuilder(base).append("a",
3L).append("b", 4L).toString());
98 }
99
100 public void testObjectArray() {
101     Object[] array = new Object[] {null, base, new int[] {3, 6}};
102     assertEquals(baseStr + "[[<null>, 5, [3, 6]]]", new
ToStringBuilder(base).append(array).toString());
103     assertEquals(baseStr + "[[<null>, 5, [3, 6]]]", new
ToStringBuilder(base).append((Object) array).toString());
104     array = null;
105     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append(array).toString());
106     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append((Object) array).toString());
107 }
108
109 public void testLongArray() {
110     long[] array = new long[] {1, 2, -3, 4};
111     assertEquals(baseStr + "[[1, 2, -3, 4]]", new
ToStringBuilder(base).append(array).toString());
112     assertEquals(baseStr + "[[1, 2, -3, 4]]", new
ToStringBuilder(base).append((Object) array).toString());
113     array = null;
114     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append(array).toString());
115     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append((Object) array).toString());
116 }
117
118 public void testLongArrayArray() {
119     long[][] array = new long[][] {{1, 2}, null, {5}};
120     assertEquals(baseStr + "[[[1, 2], <null>, [5]]]", new
ToStringBuilder(base).append(array).toString());
121     assertEquals(baseStr + "[[[1, 2], <null>, [5]]]", new
ToStringBuilder(base).append((Object) array).toString());
122     array = null;
123     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append(array).toString());
124     assertEquals(baseStr + "[<null>]", new
ToStringBuilder(base).append((Object) array).toString());
125 }
126
127 }
128

```

```

138     assertEquals(baseStr + "[3]", new
ToStringBuilder(base).append(3L).toString());
139     assertEquals(baseStr + "[a=3]", new ToStringBuilder(base).append("a",
3L).toString());
140     assertEquals(baseStr + "[a=3,b=4]", new ToStringBuilder(base).append("a",
3L).append("b", 4L).toString());
141 }
142
143 public void testObjectArray() {
144     Object[] array = new Object[] {null, base, new int[] {3, 6}};
145     assertEquals(baseStr + "[[%NULL%, 5, [3, 6]]]", new
ToStringBuilder(base).append(array).toString());
146     assertEquals(baseStr + "[[%NULL%, 5, [3, 6]]]", new
ToStringBuilder(base).append((Object) array).toString());
147     array = null;
148     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append(array).toString());
149     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append((Object) array).toString());
150 }
151
152 public void testLongArray() {
153     long[] array = new long[] {1, 2, -3, 4};
154     assertEquals(baseStr + "[[1, 2, -3, 4]]", new
ToStringBuilder(base).append(array).toString());
155     assertEquals(baseStr + "[[1, 2, -3, 4]]", new
ToStringBuilder(base).append((Object) array).toString());
156     array = null;
157     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append(array).toString());
158     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append((Object) array).toString());
159 }
160
161 public void testLongArrayArray() {
162     long[][] array = new long[][] {{1, 2}, null, {5}};
163     assertEquals(baseStr + "[[[1, 2], %NULL%, [5]]]", new
ToStringBuilder(base).append(array).toString());
164     assertEquals(baseStr + "[[[1, 2], %NULL%, [5]]]", new
ToStringBuilder(base).append((Object) array).toString());
165     array = null;
166     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append(array).toString());
167     assertEquals(baseStr + "[%NULL%]", new
ToStringBuilder(base).append((Object) array).toString());
168 }
169
170 }
171

```