

## MarkersDumpingStrategy\_1.java – MarkersSnapshotDumpingStrategy\_1.java

```

1/*****
2 * Copyright (c) 2002, 2005 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM - Initial API and implementation
10 *****/
11 package org.eclipse.core.tools.resources.metadata;
12
13 import java.io.DataInputStream;
14 import java.io.IOException;
15 import java.util.ArrayList;
16 import java.util.List;
17 import org.eclipse.core.tools.metadata.*;
18
19 /**
20 * A strategy for reading .markers files version 1. Layout:
21 * <pre>
22 * SAVE_FILE -> VERSION_ID RESOURCE+
23 * VERSION_ID ->
24 * RESOURCE -> RESOURCE_PATH MARKERS_SIZE MARKER*
25
26 * RESOURCE_PATH -> String
27 * MARKERS_SIZE -> int
28 * MARKER -> MARKER_ID TYPE ATTRIBUTES_SIZE ATTRIBUTE*
29 * MARKER_ID -> long
30 * TYPE -> INDEX | QNAME
31 * INDEX -> int int
32 * QNAME -> int String
33 * ATTRIBUTES_SIZE -> int
34 * ATTRIBUTE -> ATTRIBUTE_KEY ATTRIBUTE_VALUE
35 * ATTRIBUTE_KEY -> String
36 * ATTRIBUTE_VALUE -> INTEGER_VALUE | BOOLEAN_VALUE | STRING_VALUE | NULL_VALUE
37 * INTEGER_VALUE -> int int
38 * BOOLEAN_VALUE -> int boolean
39 * STRING_VALUE -> int String
40 * NULL_VALUE -> int
41 * </pre>
42 */
43 public class MarkersDumpingStrategy_1 implements IStringDumpingStrategy {
44     /**
45      * @see
46      * org.eclipse.core.tools.resources.metadata.IStringDumpingStrategy#dumpStringContents
47      * (DataInputStream)
48      */
49     public String dumpStringContents(DataInputStream dataInput) throws
50     IOException, DumpException {
51         StringBuffer contents = new StringBuffer();
52
53         List markerTypes = new ArrayList();
54         while (dataInput.available() > 0) {
55             String resourceName = dataInput.readUTF();

```

```

1/*****
2 * Copyright (c) 2002, 2005 IBM Corporation and others.
3 * All rights reserved. This program and the accompanying materials
4 * are made available under the terms of the Eclipse Public License v1.0
5 * which accompanies this distribution, and is available at
6 * http://www.eclipse.org/legal/epl-v10.html
7 *
8 * Contributors:
9 *   IBM - Initial API and implementation
10 *****/
11 package org.eclipse.core.tools.resources.metadata;
12
13 import java.io.DataInputStream;
14 import java.io.IOException;
15 import java.util.ArrayList;
16 import java.util.List;
17 import org.eclipse.core.tools.metadata.*;
18
19 /**
20 * A strategy for reading .markers.snap files version 1. Layout:
21 * <pre>
22 *
23 * SNAP_FILE -> [VERSION_ID RESOURCE]*
24 * VERSION_ID -> int (used for backwards compatibility)
25 * RESOURCE -> RESOURCE_PATH MARKER_SIZE MARKER+
26 * RESOURCE_PATH -> String
27 * MARKER_SIZE -> int
28 * MARKER -> MARKER_ID TYPE ATTRIBUTES_SIZE ATTRIBUTE*
29 * MARKER_ID -> long
30 * TYPE -> INDEX | QNAME
31 * INDEX -> byte int
32 * QNAME -> byte String
33 * ATTRIBUTES_SIZE -> short
34 * ATTRIBUTE -> ATTRIBUTE_KEY ATTRIBUTE_VALUE
35 * ATTRIBUTE_KEY -> String
36 * ATTRIBUTE_VALUE -> BOOLEAN_VALUE | INTEGER_VALUE | STRING_VALUE | NULL_VALUE
37 * BOOLEAN_VALUE -> byte boolean
38 * INTEGER_VALUE -> byte int
39 * STRING_VALUE -> byte String
40 * NULL_VALUE -> byte
41 * </pre>
42 */
43 class MarkersSnapshotDumpingStrategy_1 implements IStringDumpingStrategy {
44     /**
45      * @see
46      * org.eclipse.core.tools.resources.metadata.IStringDumpingStrategy#dumpStringContents
47      * (DataInputStream)
48      */
49     public String dumpStringContents(DataInputStream input) throws IOException,
50     DumpException {
51         StringBuffer contents = new StringBuffer();
52         DataInputStream dataInput = new DataInputStream(input);
53         List markerTypes = new ArrayList();
54         String resourceName = dataInput.readUTF();
55         contents.append("Resource: "); //$NON-NLS-1$

```

## MarkersDumpingStrategy\_1.java - MarkersSnapshotDumpingStrategy\_1.java

```

52     contents.append("Resource: "); //$NON-NLS-1$
53     contents.append(resourceName);
54     contents.append('¥n');
55     dumpMarkers(dataInput, contents, markerTypes);
56     contents.append('¥n');
57 }
58 return contents.toString();
59 }
60
61 private void dumpMarkers(DataInputStream input, StringBuffer contents, List
markerTypes) throws IOException, DumpException {
62     int markersSize = input.readInt();
63     contents.append("Markers ["); //$NON-NLS-1$
64     contents.append(markersSize);
65     contents.append("]:"); //$NON-NLS-1$
66     contents.append('¥n');
67     for (int i = 0; i < markersSize; i++) {
68         contents.append("ID: "); //$NON-NLS-1$
69         contents.append(input.readLong());
70         contents.append('¥n');
71         dumpMarkerType(input, contents, markerTypes);
72         dumpAttributes(input, contents);
73     }
74 }
75
76
77 private void dumpAttributes(DataInputStream input, StringBuffer contents)
throws IOException, DumpException {
78     int attributesSize = input.readInt();
79     contents.append("Attributes ["); //$NON-NLS-1$
80     contents.append(attributesSize);
81     contents.append("]:"); //$NON-NLS-1$
82     contents.append('¥n');
83     for (int j = 0; j < attributesSize; j++) {
84         contents.append(input.readUTF());
85         int type = input.readInt();
86         Object value = null;
87         switch (type) {
88             case MarkersDumper.ATTRIBUTE_INTEGER :
89                 value = new Integer(input.readInt());
90                 break;
91             case MarkersDumper.ATTRIBUTE_BOOLEAN :
92                 value = input.readBoolean() ? Boolean.TRUE : Boolean.FALSE;
93                 break;
94             case MarkersDumper.ATTRIBUTE_STRING :
95                 value = "¥" + input.readUTF() + "¥"; //$NON-NLS-1$
96                 break;
97             case MarkersDumper.ATTRIBUTE_NULL :
98                 break;
99             default :
100                 throw new PartialDumpException("Invalid marker attribute type
found: " + type, contents); //$NON-NLS-1$
101         }
102         contents.append('=');
103         contents.append(value);

```

```

54     contents.append(resourceName);
55     contents.append('¥n');
56     dumpMarkers(dataInput, contents, markerTypes);
57
58
59
60 private void dumpMarkers(DataInputStream input, StringBuffer contents, List
markerTypes) throws IOException, DumpException {
61     int markersSize = input.readInt();
62     contents.append("Markers ["); //$NON-NLS-1$
63     contents.append(markersSize);
64     contents.append("]:"); //$NON-NLS-1$
65     contents.append('¥n');
66     for (int i = 0; i < markersSize; i++) {
67         contents.append("ID: "); //$NON-NLS-1$
68         contents.append(input.readLong());
69         contents.append('¥n');
70         dumpMarkerType(input, contents, markerTypes);
71         dumpAttributes(input, contents);
72     }
73 }
74
75 private void dumpAttributes(DataInputStream input, StringBuffer contents)
throws IOException, DumpException {
76     int attributesSize = input.readShort();
77     contents.append("Attributes ["); //$NON-NLS-1$
78     contents.append(attributesSize);
79     contents.append("]:"); //$NON-NLS-1$
80     contents.append('¥n');
81     for (int j = 0; j < attributesSize; j++) {
82         contents.append(input.readUTF());
83         byte type = input.readByte();
84         Object value = null;
85         switch (type) {
86             case MarkersSnapshotDumper.ATTRIBUTE_INTEGER :
87                 value = new Integer(input.readInt());
88                 break;
89             case MarkersSnapshotDumper.ATTRIBUTE_BOOLEAN :
90                 value = input.readBoolean() ? Boolean.TRUE : Boolean.FALSE;
91                 break;
92             case MarkersSnapshotDumper.ATTRIBUTE_STRING :
93                 value = "¥" + input.readUTF() + "¥"; //$NON-NLS-1$
94                 break;
95             case MarkersSnapshotDumper.ATTRIBUTE_NULL :
96                 break;
97             default :
98                 throw new PartialDumpException("Invalid marker attribute type
found: " + type, contents); //$NON-NLS-1$
99         }
100         contents.append('=');
101         contents.append(value);

```

## MarkersDumpingStrategy\_1.java - MarkersSnapshotDumpingStrategy\_1.java

```

104         contents.append('¥n');
105     }
106 }
107
108 private void dumpMarkerType(DataInputStream input, StringBuffer contents, List
markerTypes) throws IOException, DumpException {
109     String markerType;
110     int constant = input.readInt();
111     switch (constant) {
112         case MarkersDumper.QNAME :
113             markerType = input.readUTF();
114             markerTypes.add(markerType);
115             break;
116         case MarkersDumper.INDEX :
117             markerType = (String) markerTypes.get(input.readInt());
118             break;
119         default :
120             throw new PartialDumpException("Invalid marker type constant
found: " + constant, contents); //$NON-NLS-1$
121     }
122     contents.append("Marker Type: "); //$NON-NLS-1$
123     contents.append(markerType);
124     contents.append('¥n');
125 }
126
127 /**
128  * @see
129  org.eclipse.core.tools.resources.metadata.IStringDumpingStrategy#getFormatDescripti
on()
130  */
131 public String getFormatDescription() {
132     return "Markers snapshot file version 1"; //$NON-NLS-1$
133 }
134

```

```

102         contents.append('¥n');
103     }
104 }
105
106 private void dumpMarkerType(DataInputStream input, StringBuffer contents, List
markerTypes) throws IOException, DumpException {
107     String markerType;
108     byte constant = input.readByte();
109     switch (constant) {
110         case MarkersSnapshotDumper.QNAME :
111             markerType = input.readUTF();
112             markerTypes.add(markerType);
113             break;
114         case MarkersSnapshotDumper.INDEX :
115             markerType = (String) markerTypes.get(input.readInt());
116             break;
117         default :
118             throw new PartialDumpException("Invalid marker type constant
found: " + constant, contents); //$NON-NLS-1$
119     }
120     contents.append("Marker Type: "); //$NON-NLS-1$
121     contents.append(markerType);
122     contents.append('¥n');
123 }
124
125 /**
126  * @see
127  org.eclipse.core.tools.resources.metadata.IStringDumpingStrategy#getFormatDescripti
on()
128  */
129 public String getFormatDescription() {
130     return "Markers snapshot file version 1"; //$NON-NLS-1$
131 }
132

```