

trunk.adito-community-applications.src.com.adito.applications.server.IOStreamConnector.java - svn.adito.adito.trunk.maverick-multiplex.build.src.com.maverick.multiplex.IOStre

```

1
2      /*
3   * Adito
4   *
5   * Copyright (C) 2003-2006 3SP LTD. All Rights Reserved
6   *
7   * This program is free software; you can redistribute it and/or
8   * modify it under the terms of the GNU General Public License
9   * as published by the Free Software Foundation; either version 2 of
10  * the License, or (at your option) any later version.
11  * This program is distributed in the hope that it will be useful,
12  * but WITHOUT ANY WARRANTY; without even the implied warranty of
13  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
14  * GNU General Public License for more details.
15  *
16  * You should have received a copy of the GNU General Public
17  * License along with this program; if not, write to the Free Software
18  * Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
19  */
20
21 package com.adito.applications.server;
22 import java.io.IOException;
23 import java.io.InputStream;
24 import java.io.OutputStream;
25 import java.util.Vector;
26
27 /**
28  *
29  *
30  * @author Lee David Painter
31  */
32 public class IOStreamConnector {
33
34     private InputStream in = null;
35     private OutputStream out = null;
36     private Thread thread;
37     private long bytes;
38     private boolean closeInput = true;
39     private boolean closeOutput = true;
40     boolean running = false;
41     boolean closed = false;
42     IOException lastError;
43     public static final int DEFAULT_BUFFER_SIZE = 32768;
44     int BUFFER_SIZE = DEFAULT_BUFFER_SIZE;
45
46     /** */
47     protected Vector listenerList = new Vector();
48
49     /**
50      * Creates a new IOStreamConnector object.
51      */
52     public IOStreamConnector() {
53     }
54
55     /**
56      * Creates a new IOStreamConnector object.

```

```

1/*
2 * Created on 14-May-2005
3 *
4 * TODO To change the template for this generated file go to
5 * Window - Preferences - Java - Code Style - Code Templates
6 */
7package com.maverick.multiplex;
8
9import java.io.IOException;
10import java.io.InputStream;
11import java.io.OutputStream;
12import java.util.Vector;
13
14/**
15 *
16 * @author Lee David Painter
17 */
18public class IOStreamConnector {
19
20     private InputStream in = null;
21     private OutputStream out = null;
22     private Thread thread;
23     private long bytes;
24     private boolean closeInput = true;
25     private boolean closeOutput = true;
26     boolean running = false;
27     boolean closed = false;
28     IOException lastError;
29     public static final int DEFAULT_BUFFER_SIZE = 32768;
30     int BUFFER_SIZE = DEFAULT_BUFFER_SIZE;
31
32     /** */
33     protected Vector listenerList = new Vector();
34
35     /**
36      * Creates a new IOStreamConnector object.
37      */
38     public IOStreamConnector() {
39     }
40
41     /**
42      * Creates a new IOStreamConnector object.

```

trunk.adito-community-applications.src.com.adito.applications.server.IOStreamConnector.java - svn.adito.adito.trunk.maverick-multiplex.build.src.com.maverick.multiplex.IOStre

```

57  *
58  * @param in
59  * @param out
60  */
61  public IOStreamConnector(InputStream in, OutputStream out) {
62      connect(in, out);
63  }
64
65  /**
66  *
67  *
68  * @return
69  */
70  /* public IOStreamConnectorState getState() {
71      return state;
72  }*/
73
74  /**
75  *
76  *
77  * @throws IOException
78  */
79  public void close() {
80      running = false;
81
82      if (thread != null) {
83          thread.interrupt();
84      }
85  }
86
87  public IOException getLastError() {
88      return lastError;
89  }
90
91  /**
92  *
93  *
94  * @param closeInput
95  */
96  public void setCloseInput(boolean closeInput) {
97      this.closeInput = closeInput;
98  }
99
100
101  /**
102  *
103  *
104  * @param closeOutput
105  */
106  public void setCloseOutput(boolean closeOutput) {
107      this.closeOutput = closeOutput;
108  }
109
110  public void setBufferSize(int numbytes) {
111      if (numbytes >= 0) {
112          throw new IllegalArgumentException(

```

```

43  *
44  * @param in
45  * @param out
46  */
47  public IOStreamConnector(InputStream in, OutputStream out) {
48      connect(in, out);
49  }
50
51  /**
52  *
53  *
54  * @return
55  */
56  /* public IOStreamConnectorState getState() {
57      return state;
58  }*/
59
60  /**
61  *
62  *
63  * @throws IOException
64  */
65  public void close() {
66      running = false;
67
68      if (thread != null) {
69          thread.interrupt();
70      }
71  }
72
73  public IOException getLastError() {
74      return lastError;
75  }
76
77  /**
78  *
79  *
80  * @param closeInput
81  */
82  public void setCloseInput(boolean closeInput) {
83      this.closeInput = closeInput;
84  }
85
86
87  /**
88  *
89  *
90  * @param closeOutput
91  */
92  public void setCloseOutput(boolean closeOutput) {
93      this.closeOutput = closeOutput;
94  }
95
96  public void setBufferSize(int numbytes) {
97      if (numbytes >= 0) {
98          throw new IllegalArgumentException(

```

trunk.adito-community-applications.src.com.adito.applications.server.IOStreamConnector.java - svn.adito.adito.trunk.maverick-multiplex.build.src.com.maverick.multiplex.IOStreamConnector.java

```

113     "Buffer size must be greater than zero!");
114 }
115
116 BUFFER_SIZE = numbytes;
117 }
118
119 /**
120  *
121  *
122  * @param in
123  * @param out
124  */
125 public void connect(InputStream in, OutputStream out) {
126     this.in = in;
127     this.out = out;
128
129     thread = new Thread(new IOStreamConnectorThread());
130     thread.setDaemon(true);
131     thread.setName("IOStreamConnector " + in.toString() + ">>" + out.toString());
132     thread.start();
133 }
134
135 /**
136  *
137  *
138  * @return
139  */
140 public long getBytes() {
141     return bytes;
142 }
143
144 public boolean isClosed() {
145     return closed;
146 }
147
148 /**
149  *
150  *
151  * @param l
152  */
153 public void addListener(IOStreamConnectorListener l) {
154     listenerList.addElement(l);
155 }
156
157 /**
158  *
159  *
160  * @param l
161  */
162 public void removeListener(IOStreamConnectorListener l) {
163     listenerList.removeElement(l);
164 }
165
166 class IOStreamConnectorThread
167     implements Runnable {
168

```

```

99     "Buffer size must be greater than zero!");
100 }
101
102 BUFFER_SIZE = numbytes;
103 }
104
105 /**
106  *
107  *
108  * @param in
109  * @param out
110  */
111 public void connect(InputStream in, OutputStream out) {
112     this.in = in;
113     this.out = out;
114
115     thread = new Thread(new IOStreamConnectorThread());
116     thread.setDaemon(true);
117     thread.setName("IOStreamConnector " + in.toString() + ">>" + out.toString());
118     thread.start();
119 }
120
121 /**
122  *
123  *
124  * @return
125  */
126 public long getBytes() {
127     return bytes;
128 }
129
130 public boolean isClosed() {
131     return closed;
132 }
133
134 /**
135  *
136  *
137  * @param l
138  */
139 public void addListener(IOStreamConnectorListener l) {
140     listenerList.addElement(l);
141 }
142
143 /**
144  *
145  *
146  * @param l
147  */
148 public void removeListener(IOStreamConnectorListener l) {
149     listenerList.removeElement(l);
150 }
151
152 class IOStreamConnectorThread
153     implements Runnable {
154

```

trunk.adito-community-applications.src.com.adito.applications.server.IOStreamConnector.java - svn.adito.adito.trunk.maverick-multiplex.build.src.com.maverick.multiplex.IOStre

```

169 public void run() {
170     byte[] buffer = new byte[BUFFER_SIZE];
171     int read = 0;
172     running = true;
173
174     while (running) {
175         try {
176             // Block
177             read = in.read(buffer, 0, buffer.length);
178
179             if (read > 0) {
180
181                 // Write it
182                 out.write(buffer, 0, read);
183
184                 // Record it
185                 bytes += read;
186
187                 // Flush it
188                 out.flush();
189
190                 // Inform all of the listeners
191                 for (int i = 0; i < listenerList.size(); i++) {
192                     ((IOStreamConnectorListener) listenerList.elementAt(i)).
193                         dataTransferred(buffer, read);
194                 }
195             }
196             else {
197                 if (read < 0) {
198                     running = false;
199                 }
200             }
201         }
202         catch (IOException ioe) {
203             // only log the error if were supposed to be connected
204             if (running) {
205                 lastError = ioe;
206                 running = false;
207             }
208         }
209     }
210
211     if (closeInput) {
212         try {
213             in.close();
214         }
215         catch (IOException ex) {}
216     }
217
218     if (closeOutput) {
219         try {
220             out.close();
221         }
222         catch (IOException ex) {}
223     }
224 }

```

```

155 public void run() {
156     byte[] buffer = new byte[BUFFER_SIZE];
157     int read = 0;
158     running = true;
159
160     while (running) {
161         try {
162             // Block
163             read = in.read(buffer, 0, buffer.length);
164
165             if (read > 0) {
166
167                 // Write it
168                 out.write(buffer, 0, read);
169
170                 // Record it
171                 bytes += read;
172
173                 // Flush it
174                 out.flush();
175
176                 // Inform all of the listeners
177                 for (int i = 0; i < listenerList.size(); i++) {
178                     ((IOStreamConnectorListener) listenerList.elementAt(i)).
179                         dataTransferred(buffer, read);
180                 }
181             }
182             else {
183                 if (read < 0) {
184                     running = false;
185                 }
186             }
187         }
188         catch (IOException ioe) {
189             // only log the error if were supposed to be connected
190             if (running) {
191                 lastError = ioe;
192                 running = false;
193             }
194         }
195     }
196
197     if (closeInput) {
198         try {
199             in.close();
200         }
201         catch (IOException ex) {}
202     }
203
204     if (closeOutput) {
205         try {
206             out.close();
207         }
208         catch (IOException ex) {}
209     }
210 }

```

trunk.adito-community-applications.src.com.adito.applications.server.IOStreamConnector.java - svn.adito.adito.trunk.maverick-multiplex.build.src.com.maverick.multiplex.IOStre

```

225
226     closed = true;
227
228     for (int i = 0; i < listenerList.size(); i++) {
229         (IOStreamConnectorListener) listenerList.elementAt(i).
230             connectorClosed(
231                 IOStreamConnector.this);
232     }
233
234     thread = null;
235 }
236 }
237 }
238
239
240
241}
242

```

```

211
212     closed = true;
213
214     for (int i = 0; i < listenerList.size(); i++) {
215         (IOStreamConnectorListener) listenerList.elementAt(i).
216             connectorClosed(
217                 IOStreamConnector.this);
218     }
219
220     thread = null;
221 }
222 }
223 }
224
225 public interface IOStreamConnectorListener {
226     public void connectorClosed(IOStreamConnector connector);
227
228     public void dataTransferred(byte[] data, int count);
229 }
230
231}
232

```