

acqliteCore.components.rudp.src.test.java.org.limewire.rudp.StubConnectObserver.java - svn.acqlite.AcqliteCore.components.nio.src.test.java.org.limewire.nio.observer.StubConnectObserver.java

```

1 package org.limewire.rudp;
2
3 import java.io.IOException;
4 import java.net.Socket;
5 import java.nio.channels.SocketChannel;
6
7 import org.limewire.nio.observer.ConnectObserver;
8
9 public class StubConnectObserver implements ConnectObserver {
10     private SocketChannel channel;
11     private Socket socket;
12     private IOException ioException;
13     private boolean shutdown;
14
15     public SocketChannel getChannel() throws IOException {
16         if(channel == null) {
17             channel = SocketChannel.open();
18             channel.configureBlocking(false);
19         }
20         return channel;
21     }
22
23     public synchronized void handleConnect(Socket socket) throws IOException {
24         this.socket = socket;
25         notify();
26     }
27
28     public void handleIOException(IOException iox) {
29         this.ioException = iox;
30     }
31
32     public synchronized void shutdown() {
33         this.shutdown = true;
34         notify();
35     }
36
37     public IOException getIOException() {
38         return ioException;
39     }
40
41     public boolean isShutdown() {
42         return shutdown;
43     }
44
45     public Socket getSocket() {
46         return socket;
47     }
48
49     public synchronized void waitForResponse(long timeout) throws Exception {
50         wait(timeout);
51     }
52
53 }
54

```

```

1 package org.limewire.nio.observer;
2
3 import java.io.IOException;
4 import java.net.Socket;
5 import java.nio.channels.SocketChannel;
6
7 public class StubConnectObserver implements ConnectObserver {
8     private volatile SocketChannel channel;
9     private volatile Socket socket;
10    private volatile IOException ioException;
11    private volatile boolean shutdown;
12
13    public SocketChannel getChannel() throws IOException {
14        if(channel == null) {
15            channel = SocketChannel.open();
16            channel.configureBlocking(false);
17        }
18        return channel;
19    }
20
21    public synchronized void handleConnect(Socket socket) throws IOException {
22        this.socket = socket;
23        notify();
24    }
25
26    public void handleIOException(IOException iox) {
27        this.ioException = iox;
28    }
29
30    public synchronized void shutdown() {
31        this.shutdown = true;
32        notify();
33    }
34
35    public IOException getIOException() {
36        return ioException;
37    }
38
39    public boolean isShutdown() {
40        return shutdown;
41    }
42
43    public Socket getSocket() {
44        return socket;
45    }
46
47    public synchronized void waitForResponse(long timeout) throws Exception {
48        wait(timeout);
49    }
50
51 }
52

```