

```

1 // joi/10/juno/GetfileCommand.java
2 /**
3 // Copyright 2003, Bill Campbell and Ethan Bolker
4
5 import java.util.*;
6 import java.io.*;
7
8 /**
9 * The Juno shell command to get a text file from the underlying
10 * operating system and copy it to a Juno text file.
11 * Usage:
12 * <pre>
13 * <pre>getfile native-filename juno-filename
14 * </pre>
15 * </pre>
16 * <pre>
17 * <pre>
18 * @version 10
19 */
20
21 class GetfileCommand extends ShellCommand
22 {
23     GetfileCommand()
24     {
25         super( "download a file to Juno",
26               "native-filename juno-filename" );
27     }
28
29
30 /**
31 * Use the getfile command to copy the content of a real
32 * file to a Juno TextFile.
33 * <p>
34 * The command has the form:
35 * <pre>
36 * get nativeFile textfile <&>
37 * @param args: the remainder of the command line.
38 * @param sh: the current shell
39 * @exception JunoException for reporting errors
40 */
41
42
43
44 public void doit( StringTokenizer args, Shell sh )
45 throws JunoException
46 {
47     if ( sh.getConsole().isRemote() ) {
48         throw( new JunoException(
49             "Get not implemented for remote consoles." ) );
50     }
51     String src;
52     String dst;
53     try {
54         src = args.nextToken();
55         dst = args.nextToken();
56     }

```

```

57     catch ( NoSuchElementException e ) {
58         throw new BadShellCommandException( this );
59     }
60     BufferedReader inStream = null;
61     Writer outStream = null;
62     try {
63         inStream = new BufferedReader( new FileReader( src ) );
64         outStream = new StringWriter();
65         String line;
66         while ((line = inStream.readLine()) != null) {
67             outStream.write( line );
68         }
69         outStream.write( '\n' );
70     }
71     new TextFile( dst, sh.getUser(),
72                   sh.getDot(), outStream.toString() );
73     outStream.write( '\n' );
74     catch ( IOException e ) {
75         throw new JunoException( "IO problem in get" );
76     }
77     finally {
78         try {
79             inStream.close();
80             outStream.close();
81         }
82         catch ( IOException e ) {}
83     }
84 }
85 }

```