

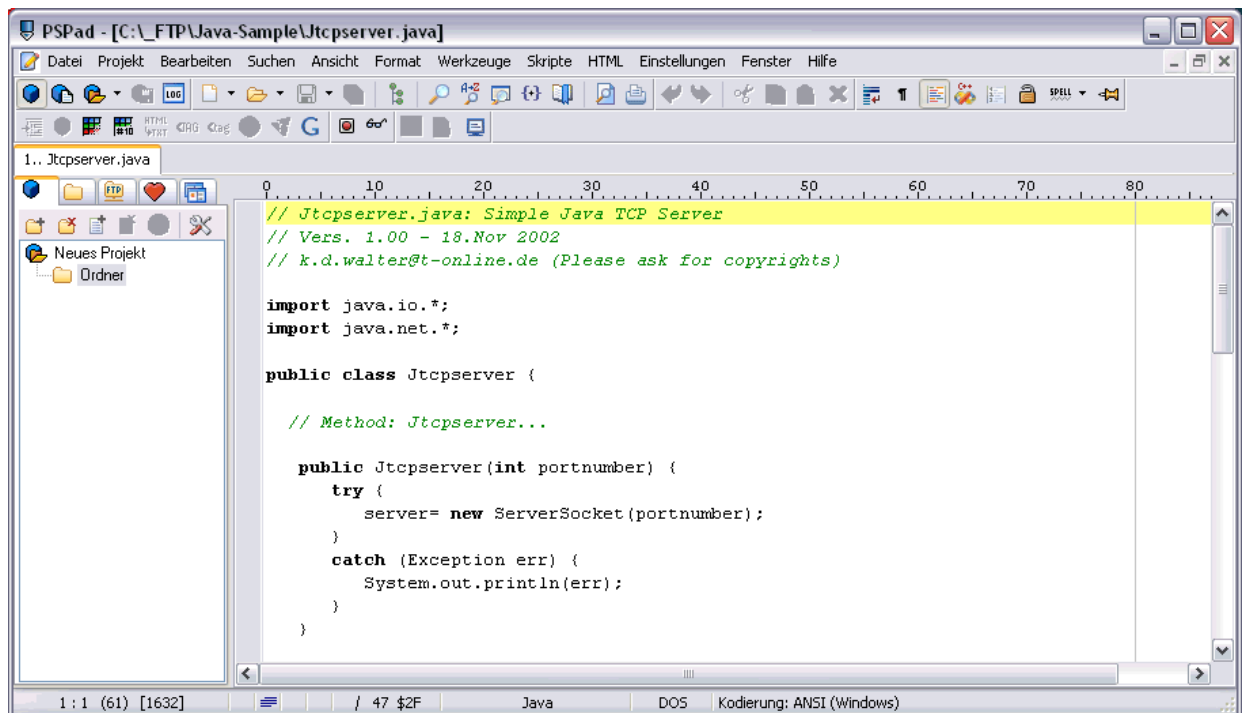
## How to write a simple TCP server for the Java Runtime Environment (JRE)

The Com/PC Embedded Gateway Linux (EGL/2) operating system comes with a pre-installed Sun Java Runtime Environment (JRE). The JRE version is 1.5.0\_09 or newer. This allows you to write Java applications and to run these apps on the Com/PC.

Before following the next steps, make sure that your development PC has a useable Sun Java Development Kit SE 5 (JDK 5). If not, please install this development environment. Please visit <http://java.sun.com/javase/downloads/index.jsp> for more information.

The following sample was made with the *Sun J2SE Development Kit 5.0 Update 13 with NetBeans IDE 5.5.1 Bundle*. The download file was *jdk-1\_5\_0\_13-nb-5\_5\_1-Win-ml.exe*. The Sun JDK 5 was installed on a Windows XP-based PC. Please see also *mHT-CPC1L-15.pdf: How to write a Hello World for the Java Runtime Environment (JRE)*.

- **1. Step:** Please download the Java TCP server sample file *Jtcpserver.java* from the Com/PC download area at [www.ssv.comm.de](http://www.ssv.comm.de) to your development PC.



```

// Jtcpserver.java: Simple Java TCP Server
// Vers. 1.00 - 18.Nov 2002
// k.d.walter@t-online.de (Please ask for copyrights)

import java.io.*;
import java.net.*;

public class Jtcpserver {

    // Method: Jtcpserver...

    public Jtcpserver(int portnumber) {
        try {
            server= new ServerSocket(portnumber);
        }
        catch (Exception err) {
            System.out.println(err);
        }
    }
}

```

- **2. Step:** Run your Java compiler and translate the file *Jtcpserver.java* to a Java class file (Java executable or Java bytecode file) on your PC.

```
javac Jtcpserver.java
```

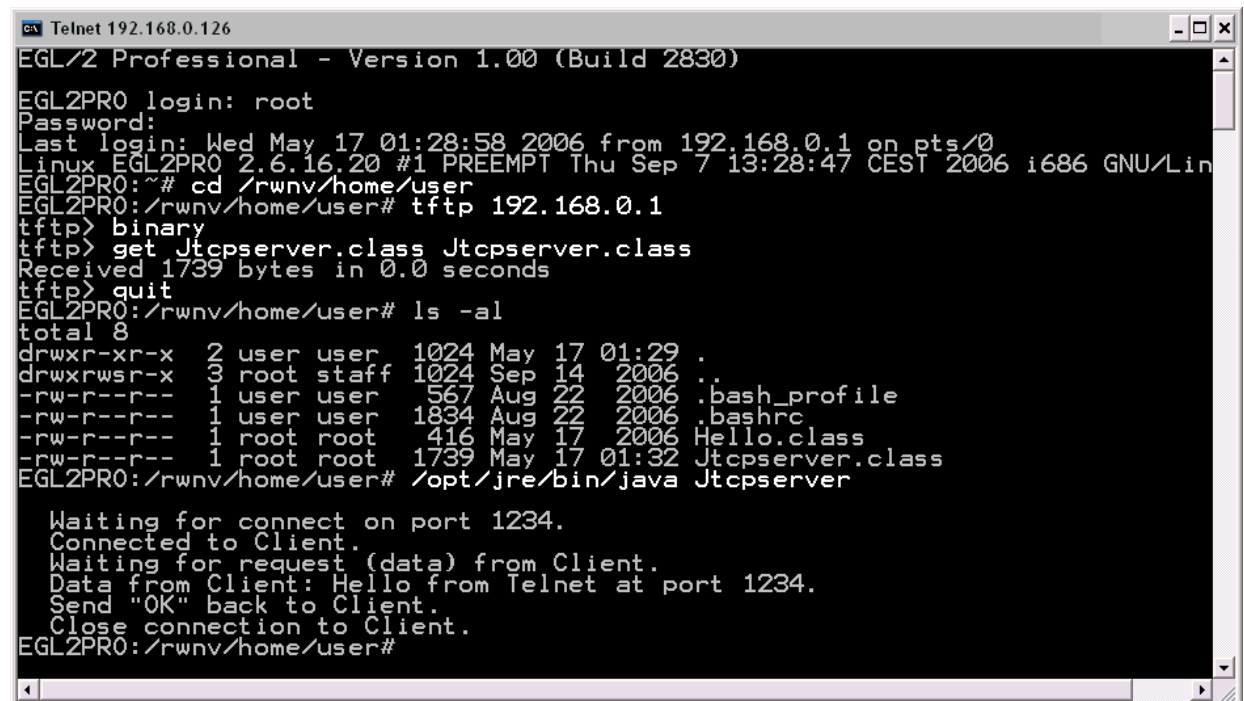
- **3. Step:** Run a SSH or Telnet client on your development PC and create a SSH or Telnet session on your Com/PC (please see also: *mHT-CPC1L-04.pdf: How to use a SSH session* and *mHT-CPC1L-05.pdf: How to use a Telnet session*). Use the user name **root** and the password **root** for your login.

- **4. Step:** Change to the Com/PC directory `/rwnv/home/user`. Transfer the Java class file `Jtcpserver.class` with FTP or TFTP from the development PC to the Com/PC EGL/2 directory `/rwnv/home/user`. The following commands for your SSH or Telnet session uses TFTP in binary mode for the file transfer:

```
cd /rwnv/home/user
tftp 192.168.0.1
binary
get Jtcpserver.class Jtcpserver.class
quit
```

Then run the Java class file on the Com/PC.

```
/opt/jre/bin/java Jtcpserver
```



```

Telnet 192.168.0.126
EGL/2 Professional - Version 1.00 (Build 2830)
EGL2PR0 login: root
Password:
Last login: Wed May 17 01:28:58 2006 from 192.168.0.1 on pts/0
Linux EGL2PR0 2.6.16.20 #1 PREEMPT Thu Sep 7 13:28:47 CEST 2006 i686 GNU/Linux
EGL2PR0:~# cd /rwnv/home/user
EGL2PR0:/rwnv/home/user# tftp 192.168.0.1
tftp> binary
tftp> get Jtcpserver.class Jtcpserver.class
Received 1739 bytes in 0.0 seconds
tftp> quit
EGL2PR0:/rwnv/home/user# ls -al
total 8
drwxr-xr-x  2 user user  1024 May 17 01:29 .
drwxrwsr-x  3 root staff 1024 Sep 14 2006 ..
-rw-r--r--  1 user user   567 Aug 22 2006 .bash_profile
-rw-r--r--  1 user user  1834 Aug 22 2006 .bashrc
-rw-r--r--  1 root root   416 May 17 2006 Hello.class
-rw-r--r--  1 root root  1739 May 17 01:32 Jtcpserver.class
EGL2PR0:/rwnv/home/user# /opt/jre/bin/java Jtcpserver
Waiting for connect on port 1234.
Connected to Client.
Waiting for request (data) from Client.
Data from Client: Hello from Telnet at port 1234.
Send "OK" back to Client.
Close connection to Client.
EGL2PR0:/rwnv/home/user#

```

- **5. Step:** Run another Telnet client on your development PC and connect this client with the TCP socket `1234`. Use the following command line:

```
telnet 192.168.0.126 1234
```

The TCP socket `1234` is used by the Com/PC Java class file `Jtcpserver.class`. This TCP server is the other side of the connection for the new Telnet client. Then enter a line of text and press enter within the Telnet client windows.

That is all.