

SV TEHS SIA

Development Tools for Java™

---

# IPVES Application Note 05:

## Telnet Server

SV TEHS SIA

## **IPJV-ES Application Note 05: Telnet Server**

---

**V 1.0**

© SV TEHS SIA

Ruses 14-24 • LV1029 • Riga • Latvia

Phone: +371-9237495 +371-9223895 • Fax: +371-7332773

Email: [info@svtehs.com](mailto:info@svtehs.com) • Web: <http://www.svtehs.com>

IPJVM and IPJV-ES are trademarks of SV TEHS SIA. ipStack, ipOS are trademarks of Ubicom, Inc. Java™ and all Java™-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other trademarks are property of their respective owners.

---



## Introduction

*IPJV-ES Development Board can be used in different applications.*

The IPJV-ES Development Board with embedded virtual machine for Java™ offers an Ethernet based connection to the Internet and numerous interface possibilities to other equipment, include serial RS-232 DTE interface, serializer module with UART, SPI, GPSI and 10BASE-T Ethernet support, 6-channel 10-bit A/D inputs, analog comparator and 16 I/O pins.

The IPJVM virtual machine for Java is a clean room implementation, that has been specially optimized to run on device with limited amount of internal memory and designed for Java™ 2 Platform, Micro Edition (J2ME™) Connected Device Configuration (CDC) Foundation Profile.

A complete development toolkit available for application development with IPJVM platform. The IPJVM platform provide system designers and software developers simple, flexible and cost-effective solution for embedded Internet application rapid development and prototyping. The platform is combination of Uvicom IP2022 Internet Processor and a Java programmable runtime environment.

The IPJV-ES Development Board based on Uvicom IP2022 Internet Processor, optimized for Internet-edge applications. It handles protocol processing in software instead of in hard-wired logic, making the whole solution more adaptable to evolving standards and allow designer to use the same solution across a wide variety of internet-edge products simply by changing the software, thereby significantly reducing nonrecurring engineering (NRE) costs.

Typical IPJV-ES applications include Includes HTTP/FTP/SMTP/SNMP/Telnet servers, PPP support on embedded UARTs, encryption, security and authentication tools, reporting and alarming via e-mail, remote monitoring, control, management and maintenance.

## Updates

New versions of the IPJV-ES software and applications can be obtained from the manufacturer's web site at:

<http://www.svtehs.com/ipjv.htm>

## Telnet Server

*Telnet Server can be used for configuration and direct hardware access.*

**T**elnet server for IPJV-ES Development Board demonstrate several possibilities for direct access to the hardware resources. It consist of the main class `TestTerm.java`, terminal class `JBTerm.java`, general command supporting class `JBTermCmd.java` and separate classes for each supporting command `tr*.java`.

### Main class `TestTerm.java`

Main class open new thread with telnet server on port 23 and call terminal class `JBTerm.java`.

### Terminal class `JBTerm.java`

Terminal class implement all server command definitions and aliases. All known to the server commands listed in the `cmdAliases` array. Necessary commands can be added or changed easily.

```
import jbvm.ip2k.*;
import java.util.*;
import java.io.*;
import java.net.*;
import java.lang.reflect.*;
//-----
public class JBTerm implements Runnable
{
public InputStream in;
public OutputStream sout;
public PrintStream out;
JBTermCmd cmdObj;
```

## TELNET SERVER

```
public String argv[];
public int argc;
String cmdCl;
public String curdir="/";
int lastch=0;
/* Command array */
static String cmdAliases[]={
    "mem", "trMem",
    "cd", "trCd",
    "dir", "trDir",
    "ls", "trDir",
    "del", "trDel",
    "rm", "trDel",
    "rmdir", "trRmdir",
    "mkdir", "trMkdir",
    "ren", "trRen",
    "bcfg", "trBcfg",
    "reset", "trReset",
    "setip", "trSetIP0",
    "setmask", "trSetMsk0",
    "setmac", "trSetMac0",
    "setgw", "trSetGw",
    "showip", "trShowIP",
    "time", "trTime",
    "setvm", "trSetVM",
    "setfw", "trSetFW",
    "ledtest", "trTestLed",
    "host", "trHost",
    "help", "",
    "quit", "",
};
```

When connecting to the IPJV Telnet Server, welcome message appears. It also prints classes, hardware and firmware versions and wait for the command:

```
String getLine() throws IOException
{
    StringBuffer bf=new StringBuffer();
    while (true)
    {
        int ch=in.read();
        if (ch=='\n' && lastch=='\r') continue;
        lastch=ch;
        if (ch<0 || ch=='\n' || ch=='\r') break;
        if (ch==8) { int ps=bf.length()-1; if (ps>=0)
bf.deleteCharAt(ps) ; out.write('\ '); continue; }
        out.print((char)ch); out.flush();
        bf.append((char)(ch));
        if (bf.length()>512) break;
    }
    return new String(bf);
}

/* */
public JBTerm(InputStream iin,OutputStream iout)
{
    in=iin;
    out=new PrintStream(iout);
    sout=iout;
}

/* */
public void run()
{
    String siv,sia;
    int adr,val;
```

## TELNET SERVER

```
try {
try {
    out.println();
    out.println("Welcome to IPJV!");
    val=HWConfig.getFWVersion();
    sia="" + Integer.toString((val>>24)&0xff) + "." +
Integer.toString((val>>16)&0xff) + "." + Integer.toString((val>>8)&0xff) +
"." + Integer.toString((val>>0)&0xff);
    out.println("ClassesVersion"+
System.getProperty("vm.classes.version"));
    out.println("FW Version "+sia);
    out.println("HW Version "+HWConfig.getHWVersion());
while (true)
    {
    cmdObj=null;
    out.print(curdir);
    out.print(">");
    sia=getLine();
    out.println("");
    if (sia.length()==0) { continue; }
    parseArgs(sia);
    if (argc==0) { continue; }
    if (argv[0].equals("help"))
        {
        if (argc>1)
            { findCmdCl(argv[1]); }
        if (cmdObj==null)
            { showAllHelp(); continue; }
        cmdObj.doHelp();
        continue;
        }
    if (argv[0].equals("quit")) break;

```

```

        findCmdCl(argv[0]);
        if (cmdObj==null)
            { out.println("-Command not found"); continue; }
        try { cmdObj.doCmd(); } catch (Throwable e) {
e.printStackTrace(out); out.println("-Exception!"); }
        }
    } catch (IOException e) { ; }
    try { in.close(); } catch (IOException e1) { ; }
    out.close();
    } catch (Throwable et1) { et1.printStackTrace(System.err);
System.err.println("-Exception!"); }
    }

```

“Help” command print short help message, “quit” command close telnet session. For any unknown command “-Command not found” message displayed.

## Command supporting class JBTermCmd.java

Command supporting class implements all necessary symbol conversions:

- One byte to hexadecimal value
- Short (16-bit) value to hexadecimal value
- Integer (32-bit) value to hexadecimal value
- One byte do decimal value
- Hexadecimal value to integer value
- Decimal value to integer value

## BCFG command

BCFG command implemented in `trBcfg.java` class. It will switch IPJV-ES board into the configuration mode. This command did not have parameters.

## CD command

CD command implemented in `trCd.java` class. It will switch current directory to the new, indicated in the command parameter.

## **DEL command**

DEL command implemented in `trDel.java` class. It will delete file, indicated in the command parameter.

## **DIR command**

DIR command implemented in `trDir.java` class. It will display directory, indicated in the command parameter. Without parameter it will display current directory.

## **HOST command**

HOST command implemented in `trHost.java` class. It will display IP address for the host name, indicated in the command parameter.

## **MEM command**

MEM command implemented in `trMem.java` class. It will display current memory status, including total memory, free memory, heap memory and PRAM memory. This command did not have parameters.

## **MKDIR command**

MKDIR command implemented in `trMkDir.java` class. It will create directory with the name, indicated in the command parameter.

## **REN command**

REN command implemented in `trRen.java` class. It will rename file from the name, indicated in the first command parameter, to the name, indicated in the second command parameter.

## **RESET command**

RESET command implemented in `trReset.java` class. It will initiate full reboot of the IPJV-ES Development Board.

## **RMDIR command**

RMDIR command implemented in `trRmDir.java` class. It will delete directory, indicated in the command parameter.

## **SETFW command**

SETFW command implemented in `trSetFW.java` class. It will set firmware image file, indicated in the command parameter, as the firmware in use.

## SETGW command

SETGW command implemented in `trSetGw.java` class. It will set host with IP address, indicated in the command parameter, as the default gateway.

## SETIP0 command

SETIP0 command implemented in `trSetIP0.java` class. It will set IP address, indicated in the command parameter, as the default IP address of the IPJV-ES Development Board.

## SETMAC0 command

SETMAC0 command implemented in `trSetMac0.java` class. It will set MAC address, indicated in the command parameter, as the default MAC address of the IPJV-ES Development Board.

## SETMSK0 command

SETMSK0 command implemented in `trSetMsk0.java` class. It will set subnet netmask, indicated in the command parameter, as the default subnet netmask of the IPJV-ES Development Board.

## SETVM command

SETVM command implemented in `trSetVM.java` class. It will set application image file, indicated in the command parameter, as the application in use.

## SHOWIP command

SHOWIP command implemented in `trShowIP.java` class. It will display all current network settings: IP address, subnet mask, MAC address and default gateway. This command did not have parameters.

## TIME command

TIME command implemented in `trTime.java` class. It will set/display current time. Accepted time format is the following: "Mon Mar 23 15:24:39 2004". Without parameters command will display current time.

## TESTLED command

TESTLED command implemented in `trTestLed.java` class. It is example of the simple digital and analog input/output. More details provided in the Application Note 06: Hardware I/O test.

# Table of Contents

<b>1. Introduction</b>	1
Updates	1
<b>2. Telnet Server</b>	2
Main class TestTerm.java	2
Terminal class JBTerm.java	2
Command supporting class JBTermCmd.java	6
BCFG command	6
CD command	6
DEL command	7
DIR command	7
HOST command	7
MEM command	7
MKDIR command	7
REM command	7
RESET command	7
RMDIR command	7
SETFW command	7
SETGW command	8
SETIP0 command	8
SETMAC0 command	8
SETMSK0 command	8
SETVM command	8
SHOWIP command	8
TIME command	8
TESTLED command	8

---