

How to write a *Hello World* 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.

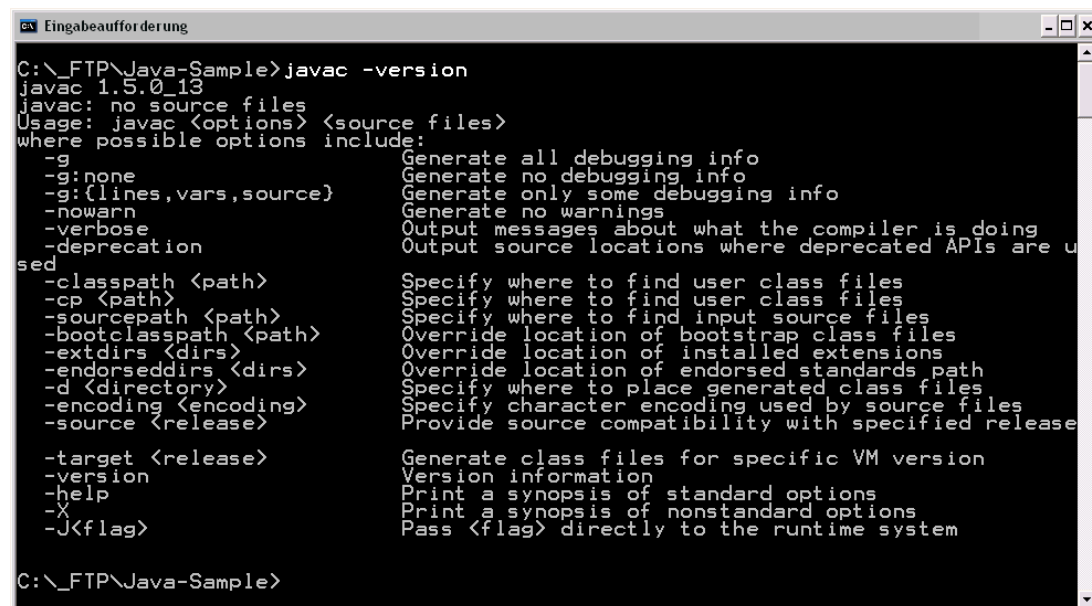
Before following the next steps, make sure that your development PC has a useable Sun Java Development Kit SE 5 (JDK 5). If not, please install this development environment. Please visit <http://java.sun.com/javase/downloads/index.jsp> for more information.

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.

- **1. Step:** Please open a command prompt window on your development PC and enter the following command:

```
javac -version
```

Make sure, that this command shows the Java compiler version. If there is any problem, please check your JDK installation. The Internet offers many helpful sites with JDK install and configuration information.



```

C:\_FTP\Java-Sample>javac -version
javac 1.5.0_13
javac: no source files
Usage: javac <options> <source files>
where possible options include:
-g          Generate all debugging info
-g:none    Generate no debugging info
-g:{lines,vars,source}  Generate only some debugging info
-nowarn    Generate no warnings
-verbose   Output messages about what the compiler is doing
-deprecation  Output source locations where deprecated APIs are used
-seed
-classpath <path>      Specify where to find user class files
-cp <path>             Specify where to find user class files
-sourcepath <path>     Specify where to find input source files
-bootclasspath <path>  Override location of bootstrap class files
-extdirs <dirs>        Override location of installed extensions
-endorseddirs <dirs>   Override location of endorsed standards path
-d <directory>        Specify where to place generated class files
-encoding <encoding>  Specify character encoding used by source files
-source <release>     Provide source compatibility with specified release

-target <release>     Generate class files for specific VM version
-version             Version information
-help               Print a synopsis of standard options
-X                 Print a synopsis of nonstandard options
-X<flag>           Pass <flag> directly to the runtime system
  
```

- **2. Step:** Run a simple editor of your choice on your PC and create the Java source code file for a simple *Hello World*. Store this file with the name *Hello.java*.

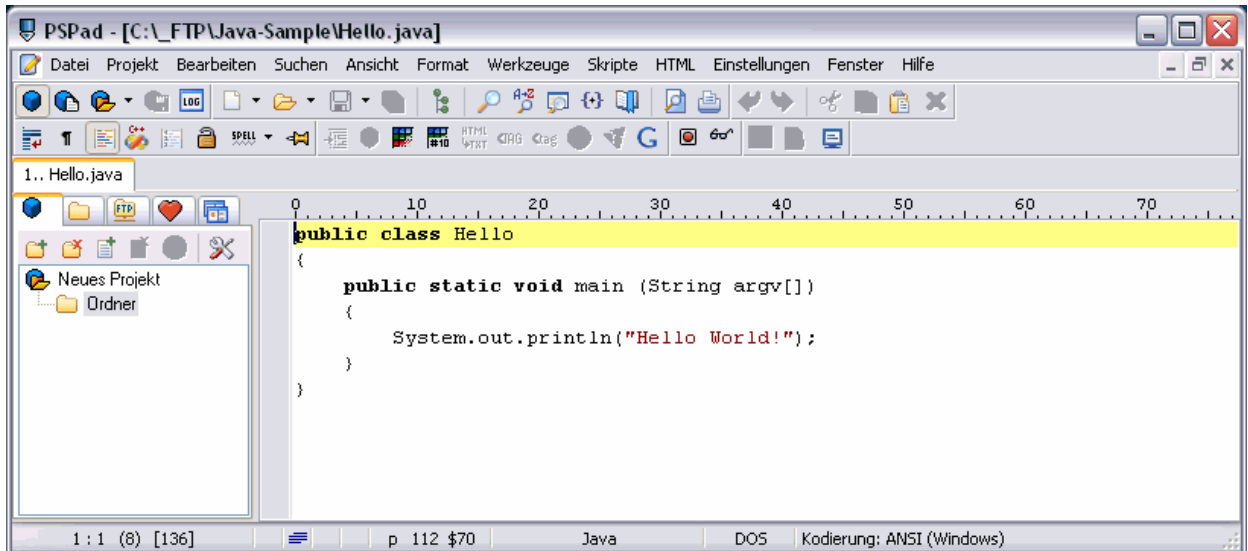
```

Public class Hello
{
    Public static void main (String argv[])
    {
  
```

```

        System.out.println("Hello World!");
    }
}

```

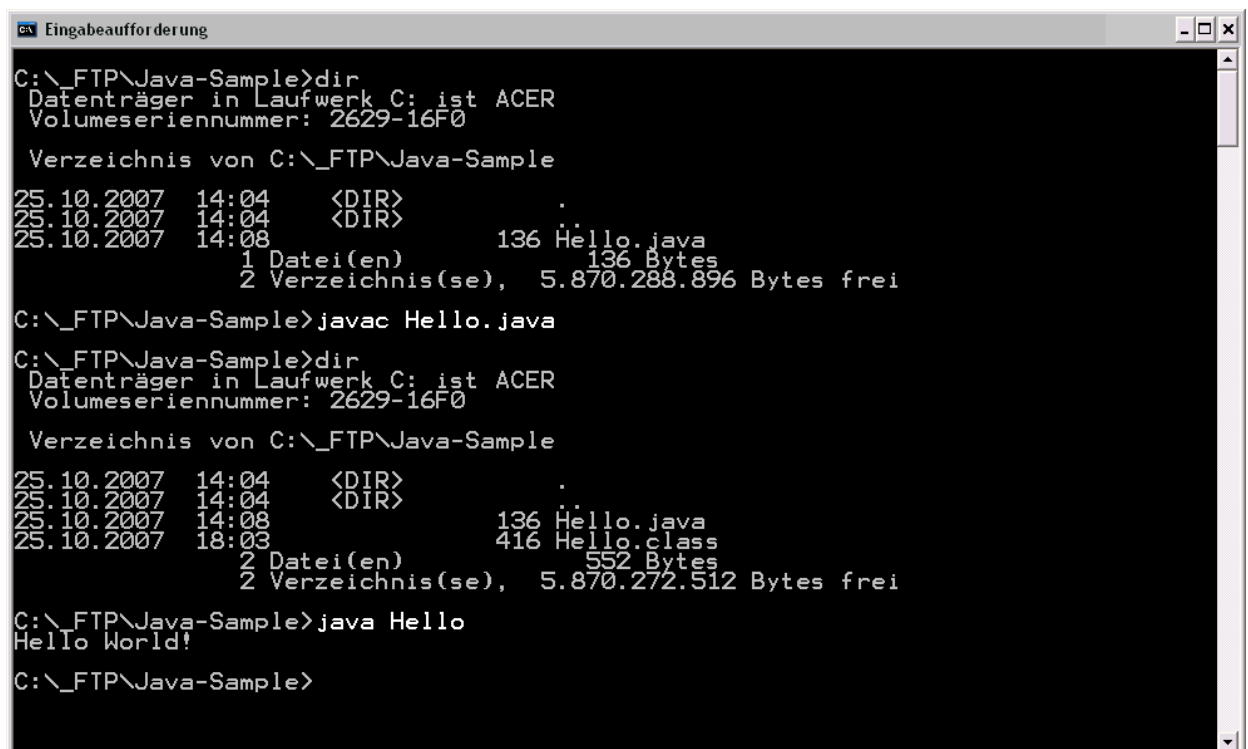


- **3. Step:** Run your Java compiler. This compiler is a part of the Sun JDK. The compiler run produces a Java class file (Java executable or Java bytecode file) on your PC. Then execute the Java class file with the help of the Java VM.

```

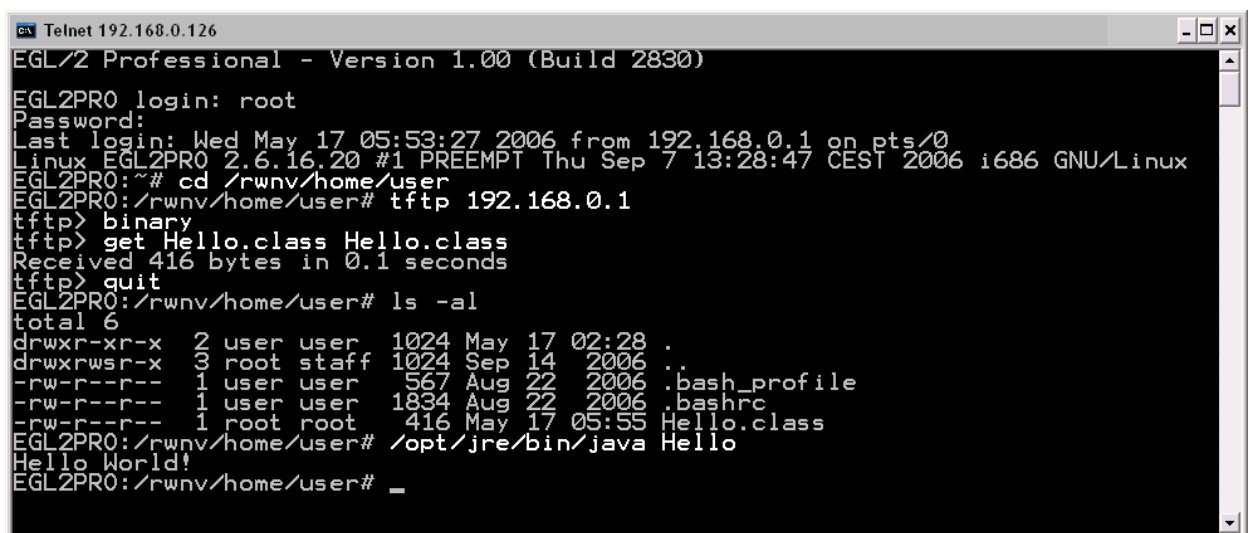
javac Hello.java
java hello

```



- **4. 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.
- **5. Step:** Change to the Com/PC directory `/rwnv/home/user`. Transfer the Java class file `Hello.class` with FTP or TFTP from the development PC to the Com/PC EGL/2 directory `/rwnv/home/user`. Then run the Java class file on the Com/PC. The following commands for your SSH or Telnet session uses TFTP:

```
cd /rwnv/home/user
tftp 192.168.0.1
binary
get Hello.class Hello.class
quit
/opt/jre/bin/java Hello
```



```
Telnet 192.168.0.126
EGL/2 Professional - Version 1.00 (Build 2830)
EGL2PR0 login: root
Password:
Last login: Wed May 17 05:53:27 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 Hello.class Hello.class
Received 416 bytes in 0.1 seconds
tftp> quit
EGL2PR0:/rwnv/home/user# ls -al
total 6
drwxr-xr-x  2 user user  1024 May 17 02:28 .
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 05:55 Hello.class
EGL2PR0:/rwnv/home/user# /opt/jre/bin/java Hello
Hello World!
EGL2PR0:/rwnv/home/user# _
```

Please note: A Java class code file is a binary file and not a text file. Make sure that your file transfer operates in binary mode. Otherwise the bytecode execution with the Com/PC Java VM produces some unclear error messages.

That is all.