**Running Java Applications as a Windows Service**

Why we need a java application as a windows service?

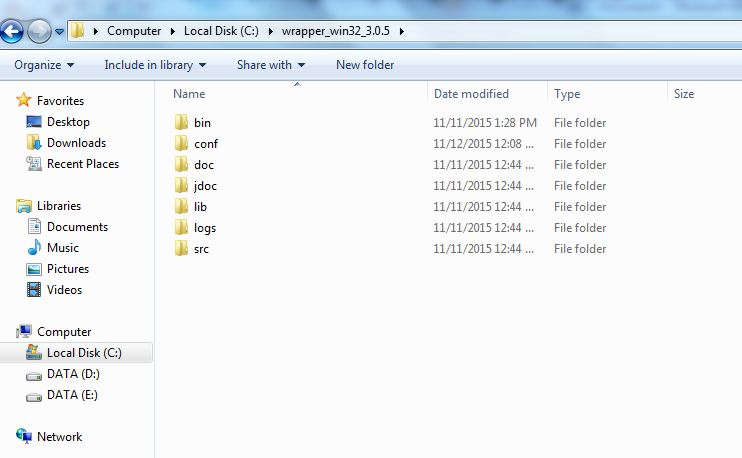
On Windows systems, Java applications are run in a command prompt.

This requires that a user account be logged on to the system at all times, and that a command prompt be open and running on the desktop. There are several drawbacks to this ranging from security, to system performance, to simply having the risk of a user pressing the wrong key on the command prompt and killing the Java application. Windows has the ability to run applications in the background as a Service to solve these problems. Services are launched on system startup and do not require that a user be logged in. This greatly increases security and overall stability of the system because it is not possible for a user to stop, start, or otherwise tamper with them unless they are an administrator.

**Steps to make a Java Application as a Windows Service are:-**

First we need to download “wrapper\_win32\_3.0.5” and extract it to ‘C’ drive(U can extract any of the disk where you want to keep the file).

In the folder ‘wrapper\_win32\_3.0.5’ you can see the sub folders



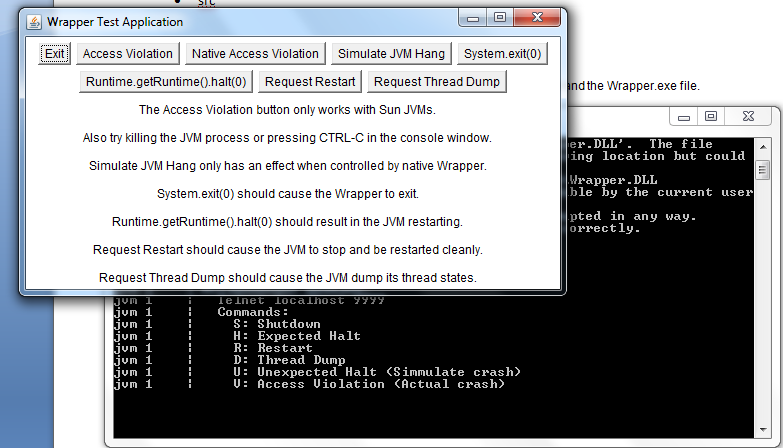
* bin
* conf
* doc
* jdoc
* lib
* logs
* src

Looking at the tree structure, we can see the below:

1. bin: contains the batch files needed to run the Java Service Wrapper and the Wrapper.exe file.
2. conf: contains the wrapper.conf file.
3. lib: contains the wrapper.dll file, wrapper.jar, and wrappertest.jar.
4. doc: contains the documentation

### The bin Folder

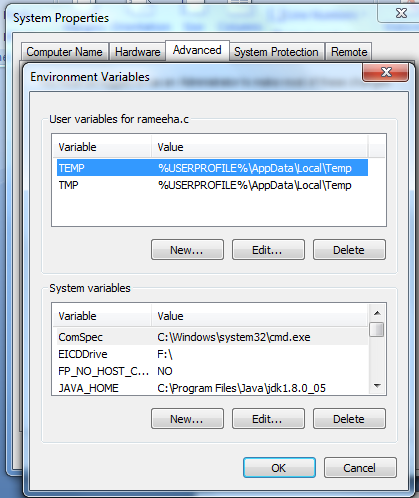
The bin folder contains several files with a .bat extension. For this example, we need only concerned with the TestWrapper.bat, installTestWrapper-NT.bat, uninstallTestWrapper-NT.bat, wrapper.exe file. These files need to be renamed to a .bat extension.  
Once you’ve done that, run the TestWrapper.bat from a command line. You should see the following results:



Now that we know the test works, let’s make one of our own Java classes work.

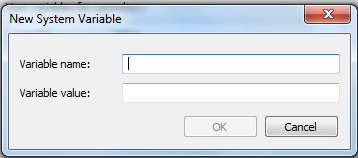
## Creating a Windows Service

* First choose java application for running as a window service. Make JAR file for that java application.
* Place the jar file to  wrapper\_win32\_3.0.5/bin directory.
* Go to my computer🡪 Right click🡪 Properties🡪Advanced system settings🡪 Environment Variables. Appear a screen like below

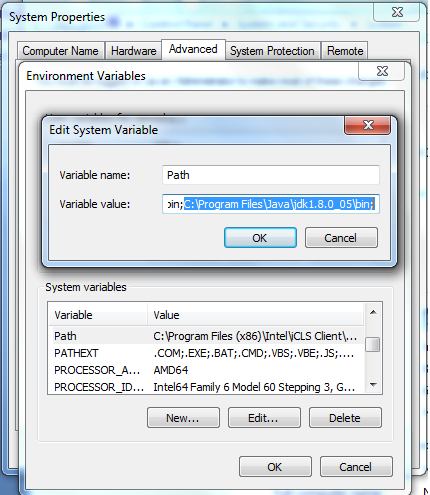


Here we can see JAVA\_HOME. If it is not present there then

Click New. Window will appear like below.



* Enter JAVA\_HOME as Variable Name.
* Enter JAVA path as Variable Value(Mostly it will be C:\Program Files\Java\jdk 1.8.0\_05)
* Then OK button will be enabled click ok.
* The JAVA\_HOME will appear on the System variables screen.
* Click on path in System variables screen and click edit. Window will appear like below

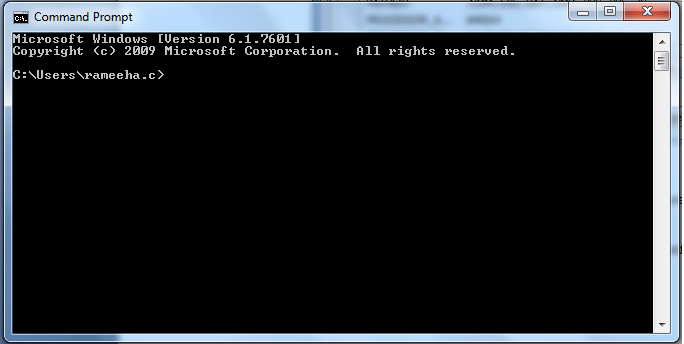


* Add ‘C:\Program Files\Java\jdk 1.8.0\_05\bin; ‘ as above picture in variable value column.
* Click Ok.

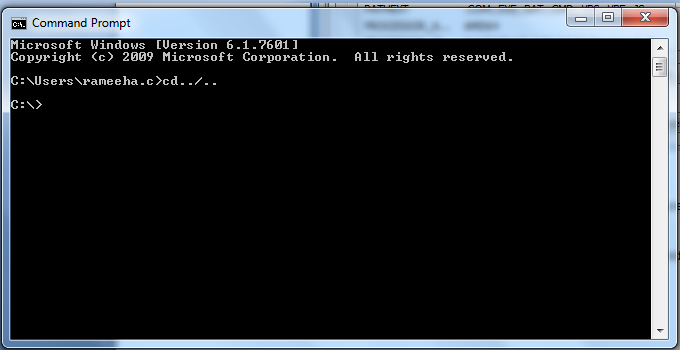
Let’s test the .jar file to make sure that everything’s ok. At a command prompt, type:

(Suppose our jar file is exeJar.jar)

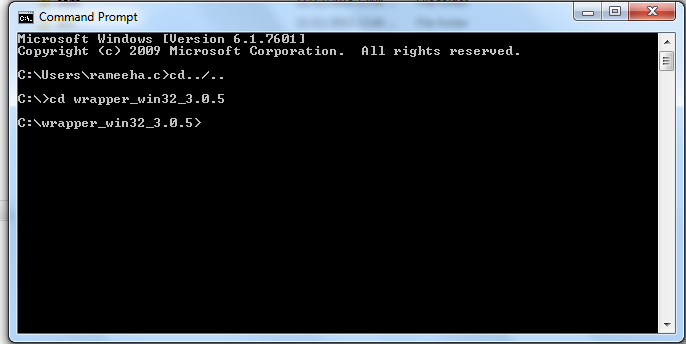
When we are opening our command prompt like below



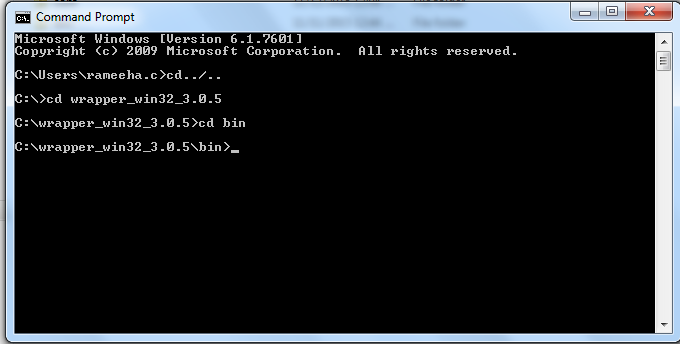
Redirect folder to C drive by entering cd../.. like below



Redirect folder to C:\wrapper\_win32\_3.0.5 by entering cd wrapper\_win32\_3.0.5 like below



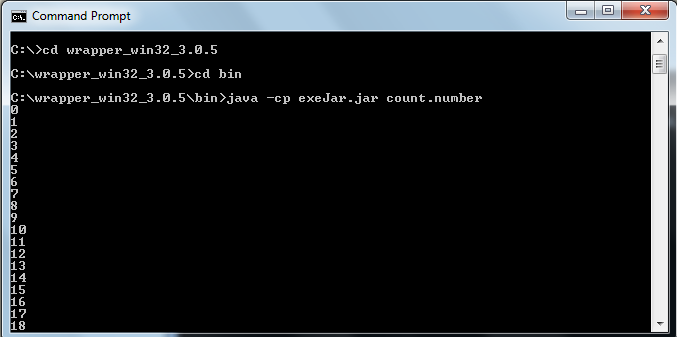
Redirect folder to C:\wrapper\_win32\_3.0.5\bin by entering cd wrapper\_win32\_3.0.5 \bin like below



Now test the exeJar.jar file to make sure that everything’s ok. At a command prompt, type:

java -cp exeJar.jar count.number

* Here exeJar.jar is the jar file for the java application
* Count is the source package name(In our case source package is POSintegerationTask)
* Number is the main class in the java application(Main class is POSClass)



We can see that our program has been running

Note: Use Ctrl-C to exit the program.

The next thing we need to do is to modify the wrapper.conf file. As mentioned earlier, this file can be found in the C:\wrapper\_win32\_3.0.5\conf directory. In this file we are going to make the following changes:

change:

wrapper.java.mainclass=org.tanukisoftware.wrapper.test.Main

to

wrapper.java.mainclass=org.tanukisoftware.wrapper.WrapperSimpleApp

add:

wrapper.java.classpath.3=exeJar.jar

below:

wrapper.java.classpath.1=../lib/wrapper.jar

wrapper.java.classpath.2=../lib/wrappertest.jar

change:

#wrapper.app.parameter.1=

to

wrapper.app.parameter.1=count.number

change:

wrapper.ntservice.name=@app.name@

wrapper.ntservice.displayname=@app.long.name@

wrapper.ntservice.description=@app.description@

to:

wrapper.ntservice.name=number

wrapper.ntservice.displayname=number

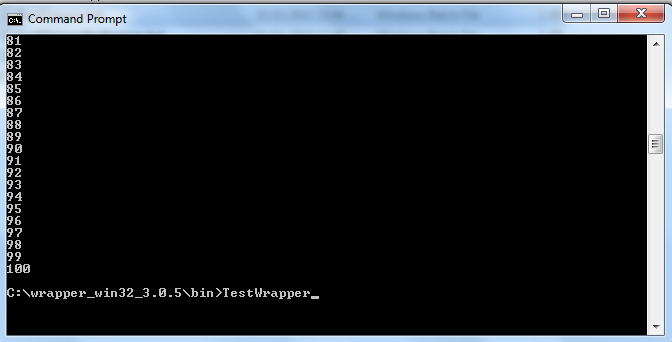
wrapper.ntservice.description=number Test

And save the file.

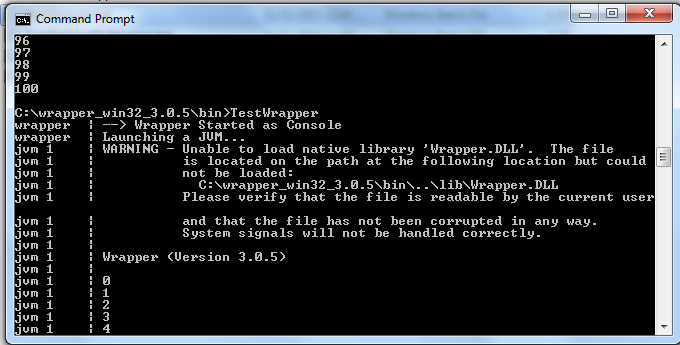
Now let’s test our creation. At the command prompt run:

C:/wrapper\_win32\_3.0.5/bin/TestWrapper

Enter like below



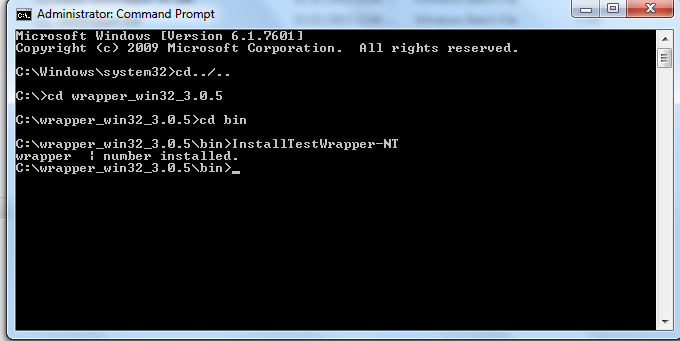
You will get an output like blow



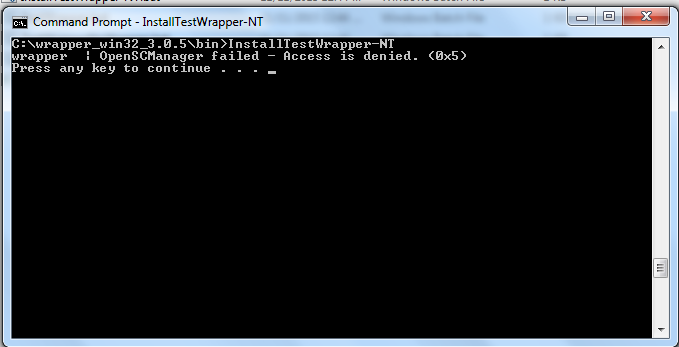
And also our program will run

Now that we’ve tested our class, let’s turn it into a full-fledged service. At the command prompt run:

C:/wrapper\_win32\_3.0.5/bin/InstallTestWrapper-NT.bat



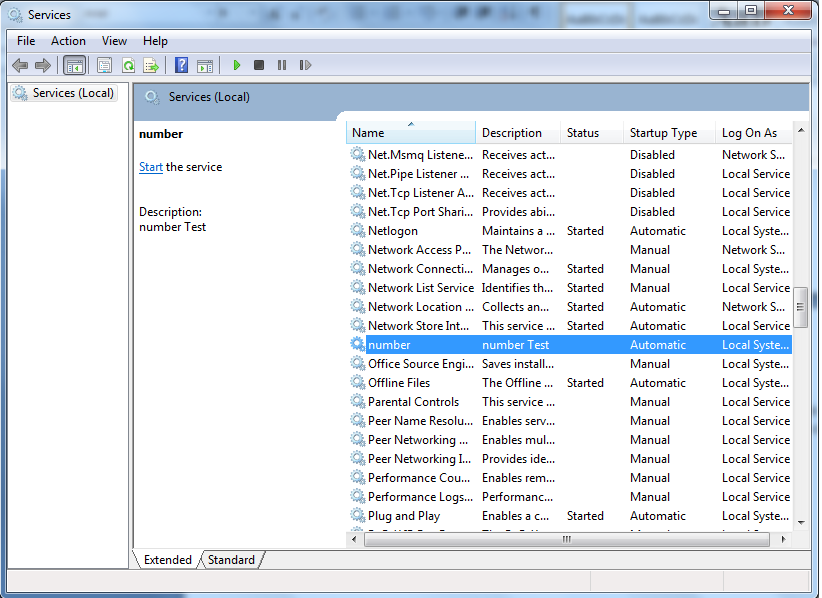
May be you will get an error like this “OpenSCManager failed – Access is denied”



To avoid this problem you need to run the command prompt as an Administrator.

* Click on start
* Click all programs
* Click on accessories
* Right click on command prompt icon
* Click on run as an administrator
* Enter the details above. You will successfully install the application as a window service

Let’s make sure that Windows recognizes the number class as a Service. Click on the “services” icon(Start🡪 services), found in the administrative tools option of the control panel. You should see a screen similar to this:



Yes!!!!!!Here is your java application as a service.

Notice that the number service is not started. To start it, you can select it and click start, or you can run the TestWrapper.bat file again. Once you’ve started the number, it will continue to run until you stop it, disable it, or run the UninstallTestWrapper-NT.bat file.

you have just built a Java based Windows service.