

Name Of Experiments : Hello World	Exp No. : NOS1
Background : Student should require a basic knowledge of Java	
Summary: After performing, this experiment student should be able to understand the proper flow of Development of J2ME based, NOS application by writing the program, debugging and deploying it on the Emulator or Target Device.	
Learning Objective: To Explain the working of NOS by writing Hello World Program.	
Target Platform: This application is tested on Nokia Series 40 Emulator and Nokia Series 40 Mobile Devices.	
<p>Procedure :</p> <p>Step 1: Open Nokia IDE for Java ME (Eclipse v2) from Nokia Asha SDK. [Refer Fig. No. no 1]</p> <p>Step 2: SDK starts IDE for you. [Refer Fig. No. no 2]</p> <p>Step 3: IDE prompts the location of workspace to store the Java Micro Edition Projects (J2ME). Browse the location and save it. [Refer Fig. No. no 3]</p> <p>Step 4: Choose File->New->MIDlet Project [Refer Fig. No. 4].We will see a new dialogue box, type the project name then click next button as shown in bottom.</p> <p>Step 5: A new dialogue box will appear [Refer Fig. No. 5], by default all the fields are filled .If we want our MIDlet and MIDlet Vendor with default names then we can click finish button on the bottom of dialogue box. Otherwise we can change it according to our convenience.</p>	

Step 6: Click the Finish Button.

Step 7: A new window will be display [Refer Fig. No. 6].

Step 8: In Navigator Panel (on left side of window), hierarchy of folder shall be created automatically by Nokia IDE for us. We can create our MIDlet in src folder by choosing src->new Java ME MIDlet [Refer Fig No. 7].

Step 9: Type the name of MIDlet and click Finish.[Refer Fig No. 8].

Step 10: A new file with name **HelloWorld.java** shall be created with a public class and 3 mandatory lifecycle methods for starting, pausing and destroying MIDlet [Refer Fig No. 9].

Step 11: Import “javax.microedition.lcdui.Form” in the uppermost section of inbuilt java file. This will import all the predefined classes and methods of javax.microedition.lcdui package (i.e Limited Connected Device User Interface).So that we can directly apply the Form and its Components in our program.

Step 12: Declare a String class “hello” to represent collection of characters in public class as “**String hello**”.

Step 13: Declare Form with name HelloWorldForm in Public Constructor HelloWorld as “**Form HelloWorldForm**”.

Step 14: Initialize the Form with the title “Hello World” as “**HelloWorldForm = new Form(“Hello World”);**” in HelloWorld() method.

Step 15: Create the object of String class as “**Hello = new String(“Hello World”);**”

Step 16: Append the String on Form by the code:
“**HelloWorldForm.append(hello);**”

Step 17: Import lcdui.Display in the very beginning of the program.

Step 18: Type the Display Command as
`Display.getDisplay(this).setCurrent(HelloWorldForm);`” in startapp() method. So that it can execute the display statement of form in the very beginning of MIDlet LifeCycle.[Refer Fig no. 10].

Step 19: Press cntrl+F11 to run the Program on Emulator. Click ok Button to save and launch. Check the output on Emulator [Refer Fig no. 11].

Step 20: To Deploy the project on real device, Refer [Deployment section](#).

Source Code	Comments
<pre>import javax.microedition.midlet.MIDlet; import javax.microedition.midlet.MIDletStateChangeException;</pre>	←import require packages

```
import javax.microedition.lcdui.Display;
```

```
import javax.microedition.lcdui.Form;
```

```
public class HelloWorld extends MIDlet {
```

```
String hello;
```

```
Form HelloWorldForm;
```

```
    public HelloWorld() {
```

```
        hello=new String("Hello World");
```

```
        HelloWorldForm=new Form("Hello World");
```

```
        HelloWorldForm.append(hello);
```

```
        // TODO Auto-generated constructor stub
```

```
    }
```

```
    protected void destroyApp(boolean arg0) throws
```

```
MIDletStateChangeException {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
    protected void pauseApp() {
```

```
        // TODO Auto-generated method stub
```

```
    }
```

```
    protected void startApp() throws
```

```
MIDletStateChangeException {
```

```
        // TODO Auto-generated method stub
```

```
        Display.getDisplay(this).setCurrent(HelloWorldForm);
```

```
    }
```

```
}
```

← Declare a String

←Declare a Form

←Create Constructor

← Add the component string on Form

←Display the Form on Emulator as well as on Real Device through inbuilt methods

Screenshots :

Fig No. 1 :

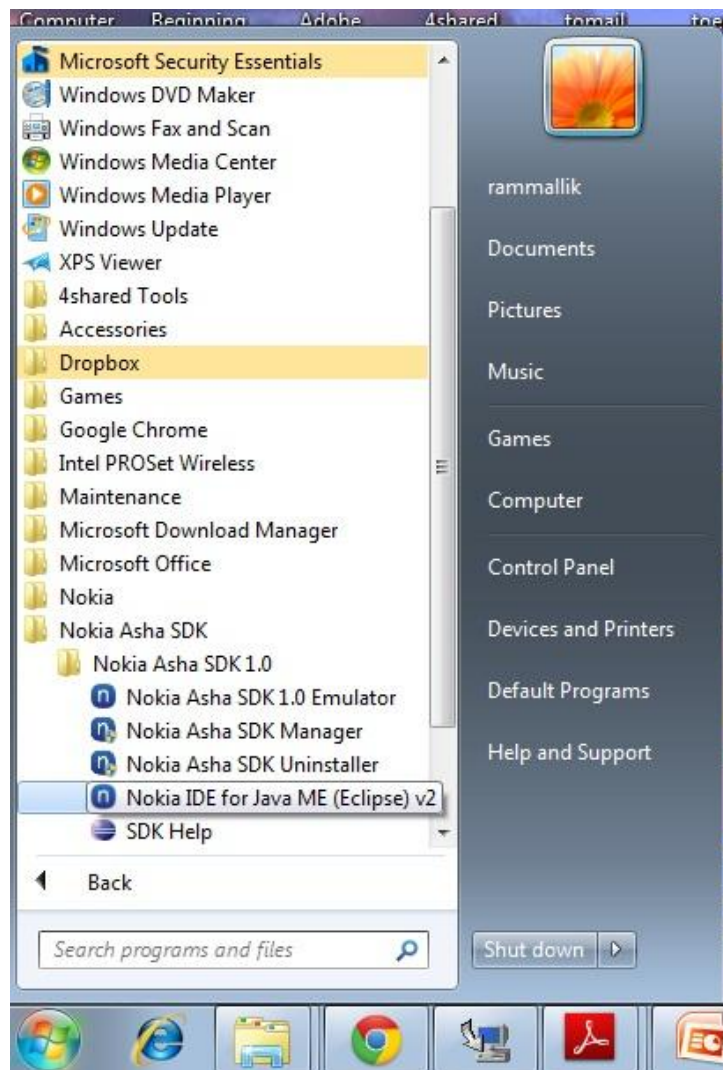


Fig No. 2:



Fig No. 3:

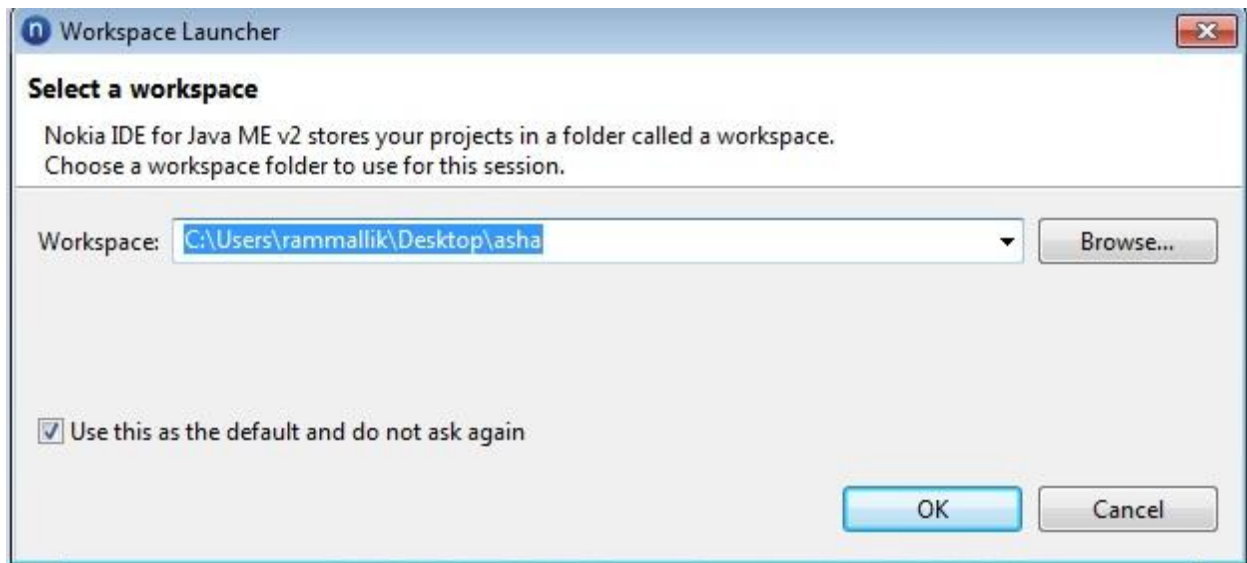


Fig No. 4:

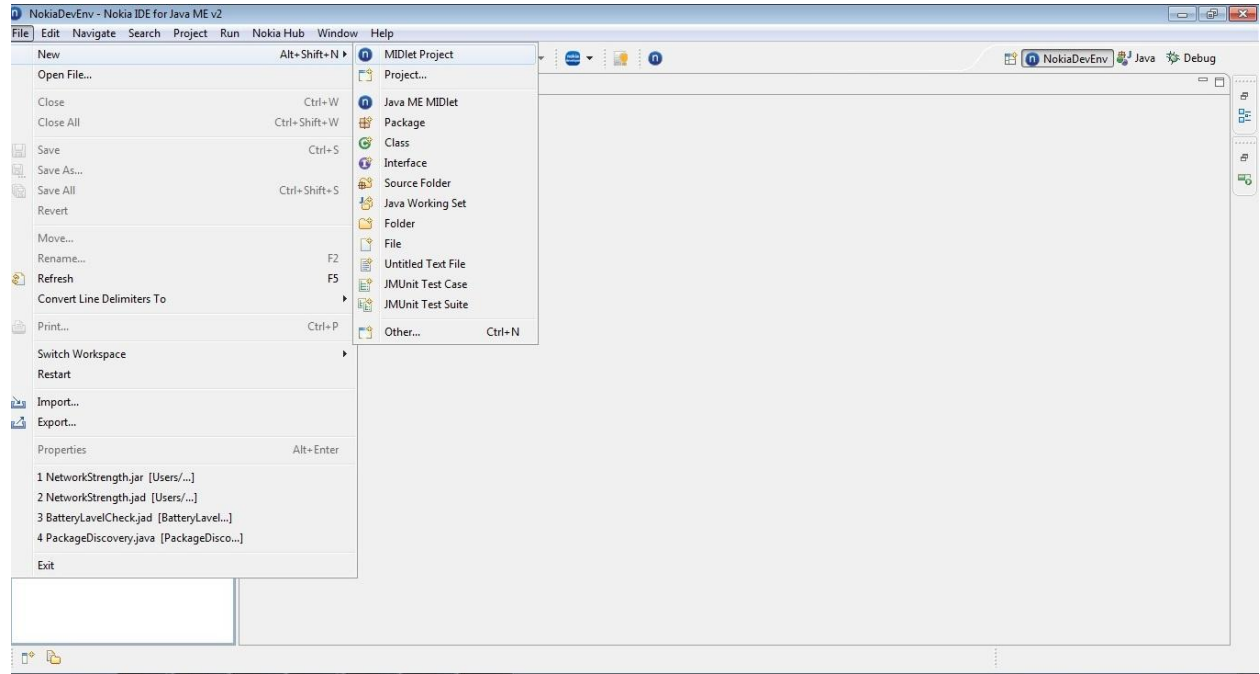


Fig No. 5

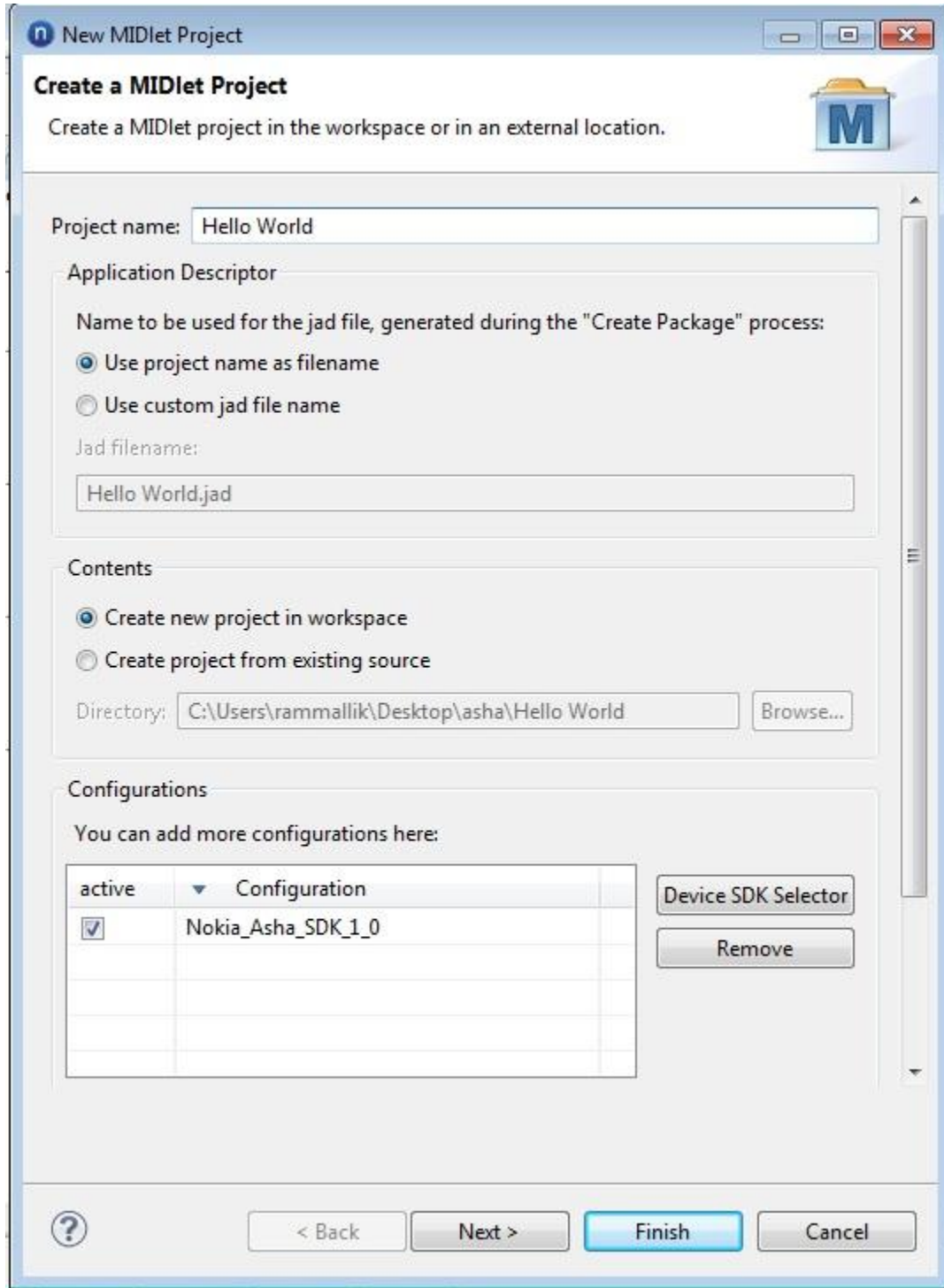


Fig No. 6:

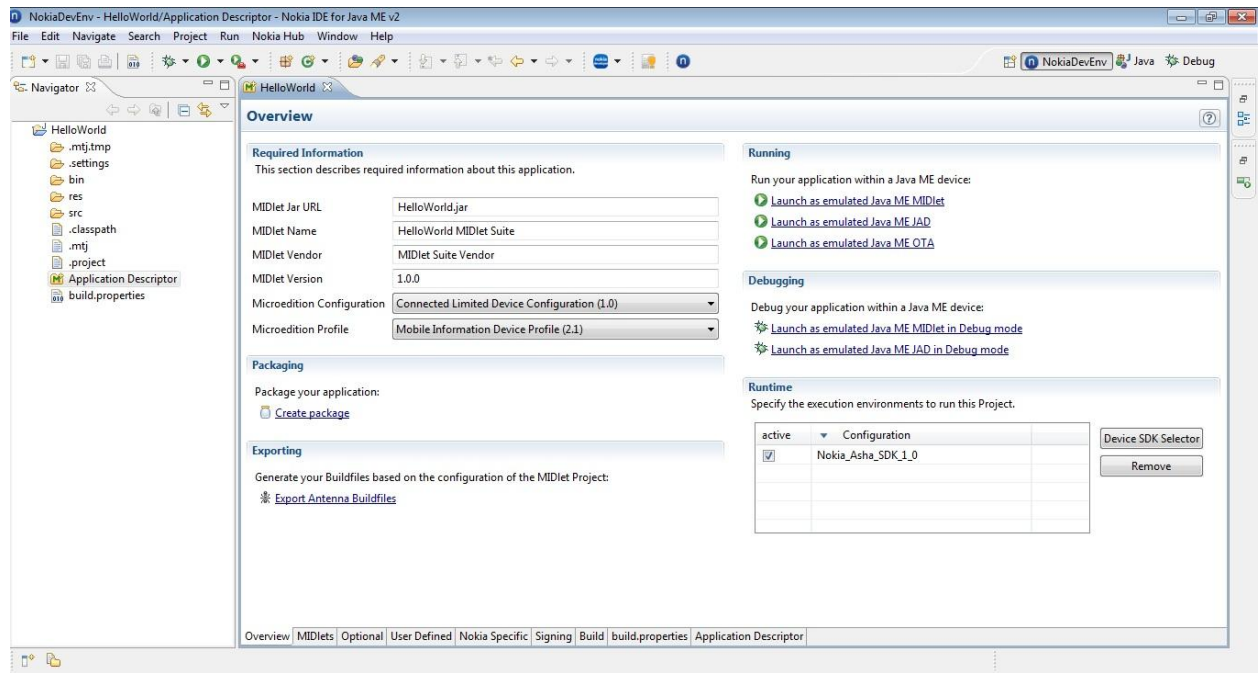


Fig No. 7:

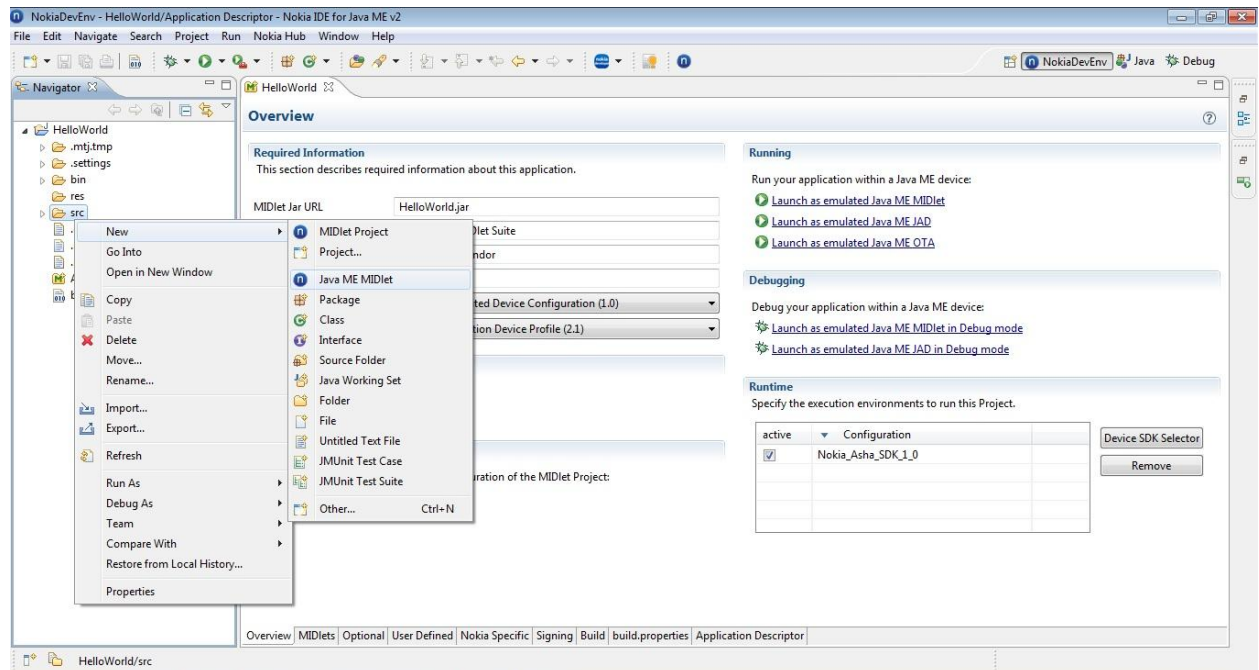


Fig No. 8:

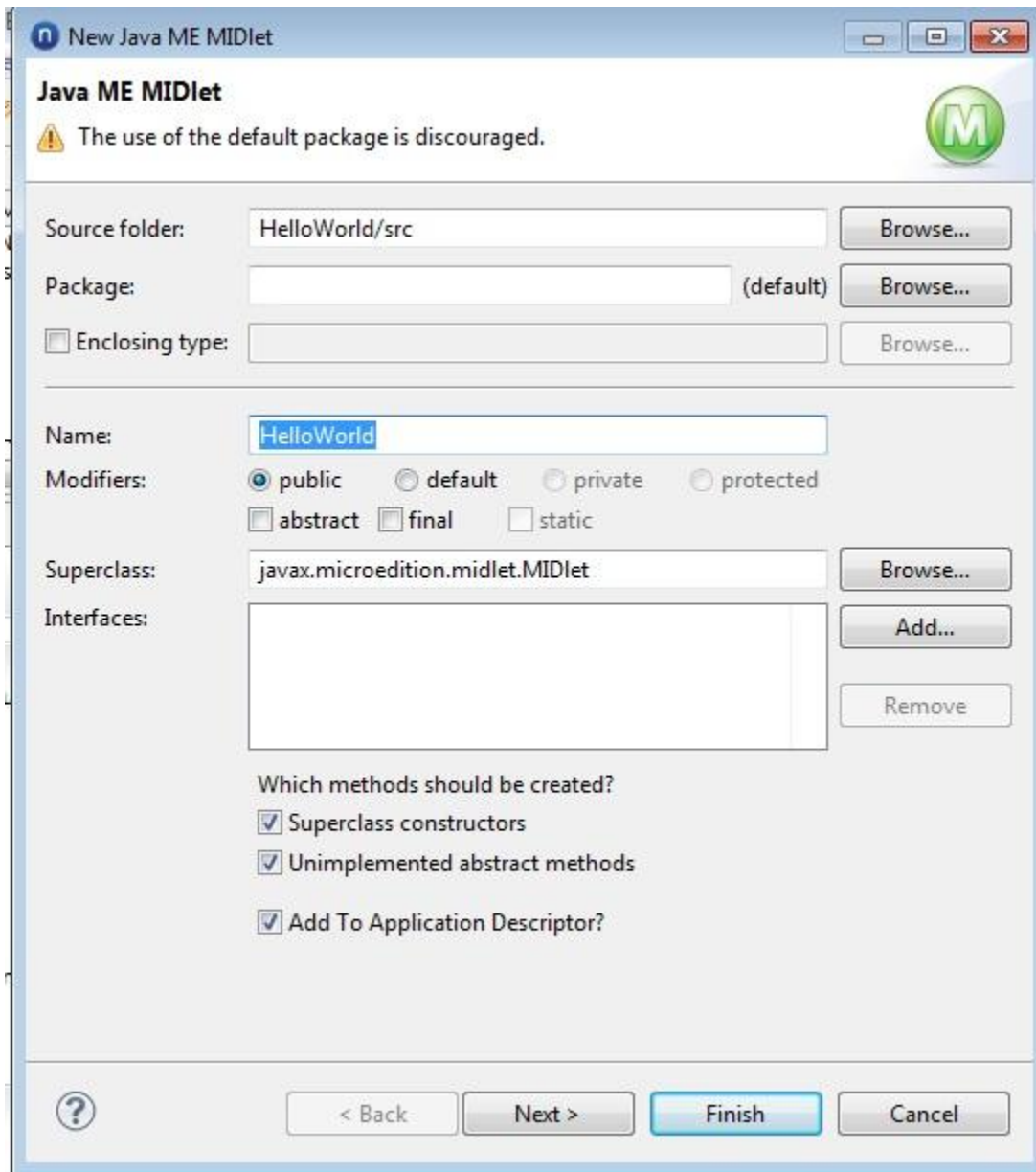


Fig No. 9:

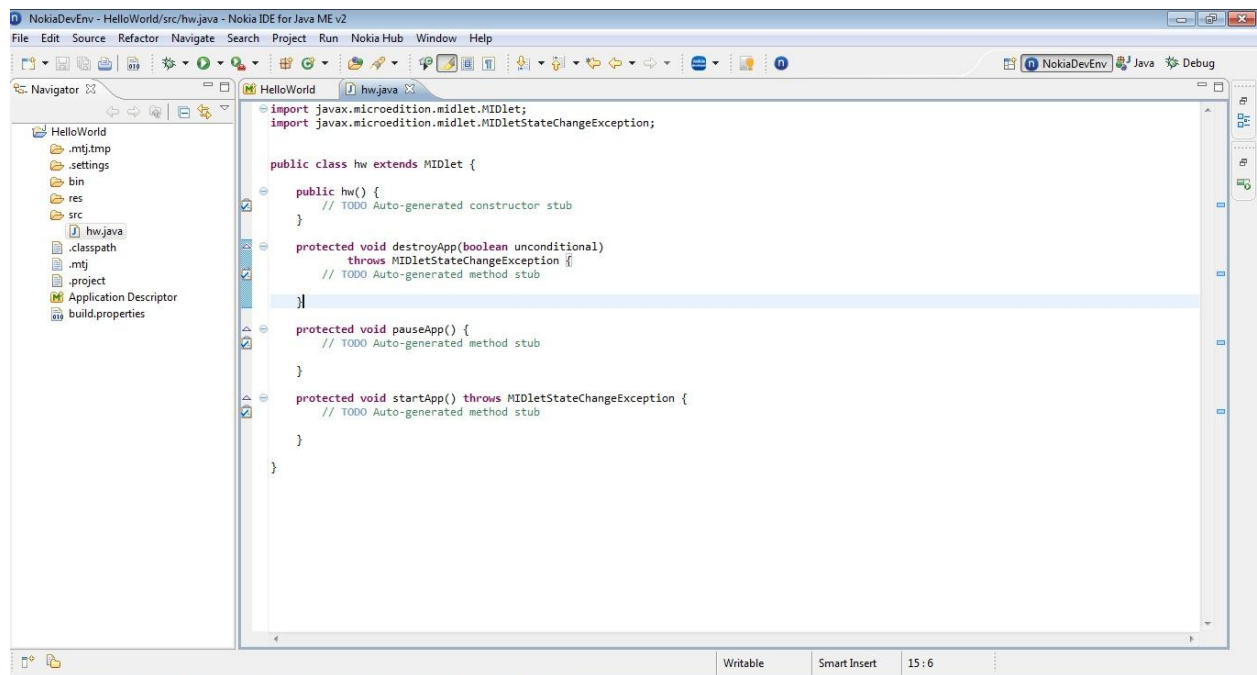


Fig No. 10:

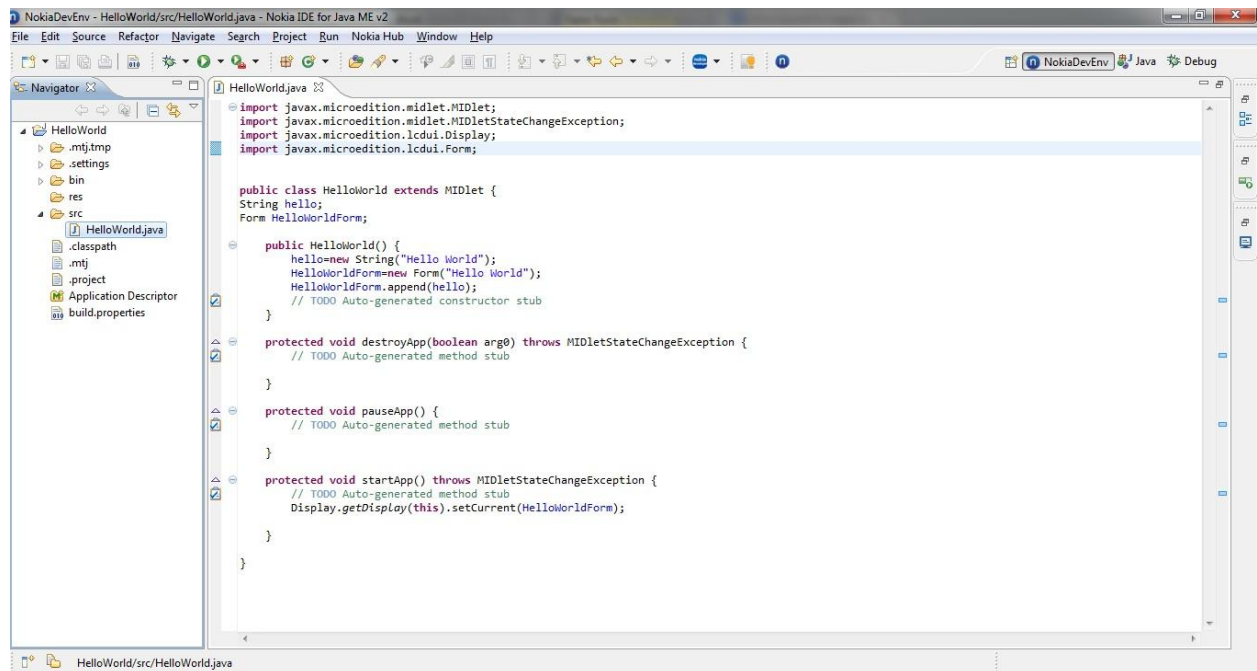


Fig No. 11



Observations: It should be noted by Developer that for any kind of J2ME application, step (1-10) need to be repeated.

Reference:

1. Download Nokia IDE -

<http://www.oracle.com/technetwork/java/javame/javamobile/download/index.html>

2. Introduction for Nokia IDE 1.0 for Java ME:

<http://www.youtube.com/watch?v=wCmUhhWk3gk>

3. Windows 7 on J2ME

<http://www.youtube.com/watch?v=EuSdVzH9j5E>

4. J2ME Tutorial : For Simple Mobile Application

<http://www.youtube.com/watch?v=fcMLVsjlxw>