Oracle FLEXCUBE Direct Banking

J2ME Phone Client Workspace Configuration
Release 12.0.3.0.0

Part No. E52543-01

April 2014

J2ME Phone Client Workspace Configuration Guide

April 2014

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001
www.oracle.com/financialservices/

Contents

1. Preface

1.1 Intended Audience

Any interested party working on the delivery of Oracle FLEXCUBE Direct Banking may read this document. The following profile of users would find this document useful:

- Application Architects
- End to End Designers
- Business Service Detailed Designers and Developers
- Implementation Partners

Specifically, however, this document is targeted at Iimplementation Partners, Customization Development Teams or Vendors providing customization, configuration and implementation services around the Oracle FLEXCUBE Direct Banking product.

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

1.3 Access to OFSS Support

https://support.us.oracle.com

1.4 Introduction

This document provides information about the development workspace setup and configurations required for Oracle FLEXCUBE Direct Banking J2ME mobile application.

1.5 Structure

This manual is organized into the following chapters:

Section 1 *Preface* gives information on the intended audience. It also lists the various chapters covered in this User Manual.

Section 2 Development gives you the information about how to set up workspace**.**

Section 3 provides information about Obfuscation

1.5 Related Information Sources

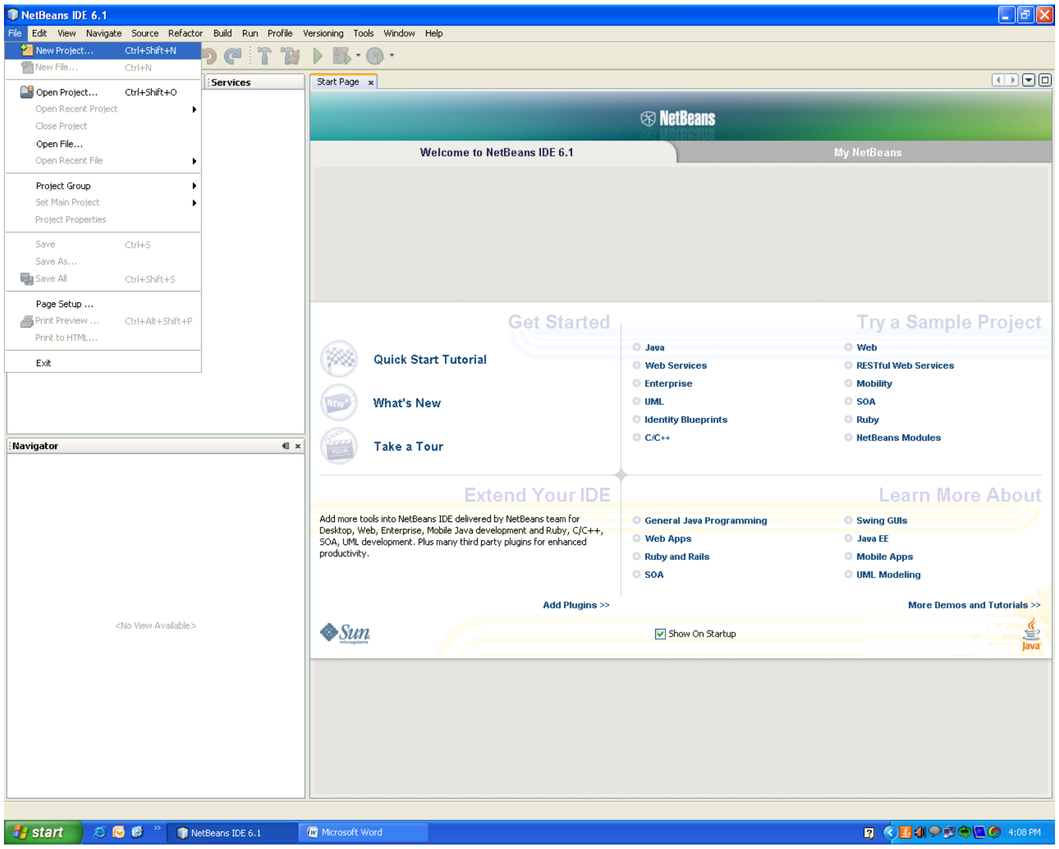
2. Development Environment

2.1 Introduction

Net Beans IDE 6.1 is used to develop and configure the project for both rich and plain version.

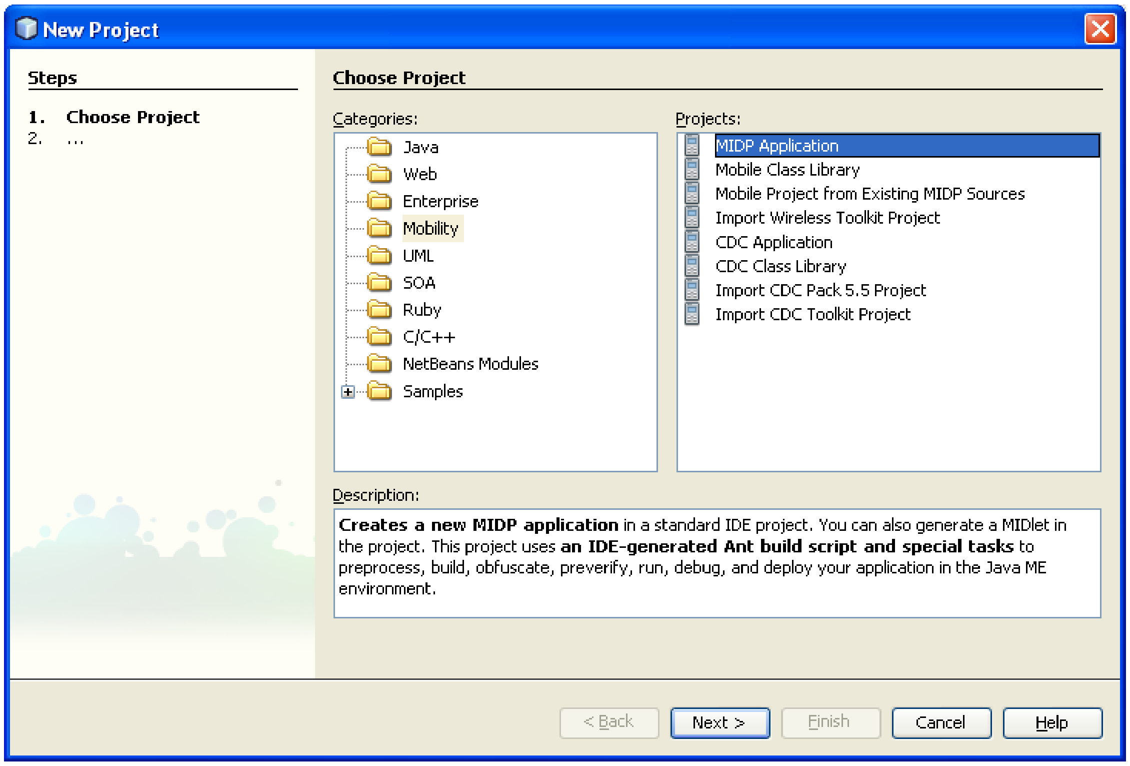
2.1.1 Steps for project set-up in netbeans

1. File -> New Project



Add more tools into NetBeans IDE delivered by NetBeans team for Desktop, Web, Enterprise, Mobile Java development and Ruby, C/C++, SOA, UML development. Plus many third party plugins for enhanced productivity.

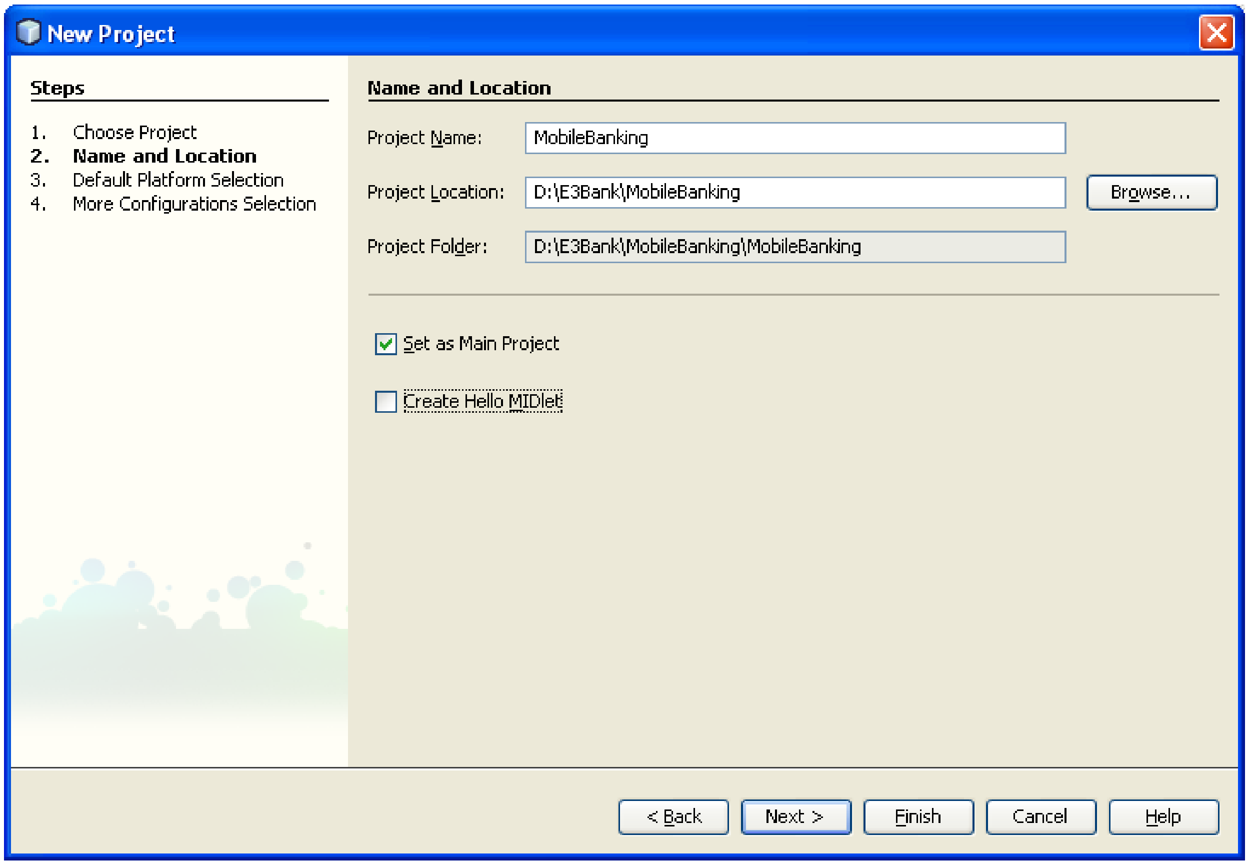
2.Select category as’mobility’ and project as ‘MIDPApplication’. Click next



Creates a new MIDP application in a standard IDE project. You can also generate a MIDlet in the project. This project uses **an IDE-generated Ant build script and special tasks** to preprocess, build, obfuscate, preverify, run, debug, and deploy your application in the Java ME environment.

3.Mention Project name and Project location. Ideally Project location should be an empty folder

Select ‘ Set as Main Project’ . Click Next


New Project

Steps
1. Choose Project
2. **Name and Location**
3. Default Platform Selection
4. More Configurations Selection

Name and Location

Project Name: MobileBanking

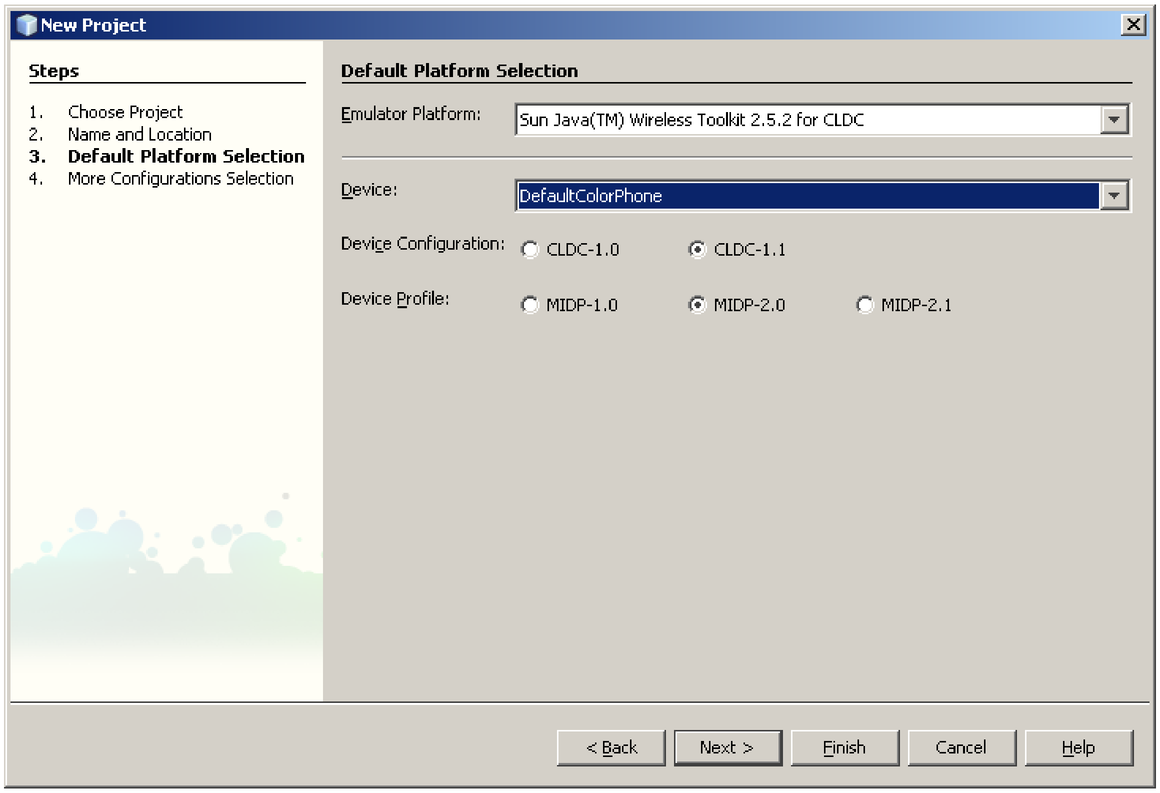
Project Location: D:\E3Bank\MobileBanking

Project Folder: D:\E3Bank\MobileBanking\MobileBanking

[✓] Set as Main Project

[ ] Create Hello MIDlet

4. Select CLDC 1.1 and MIDP 2.0. Click Next


New Project

Steps

1. Choose Project
2. Name and Location
3. **Default Platform Selection**
4. More Configurations Selection

Default Platform Selection

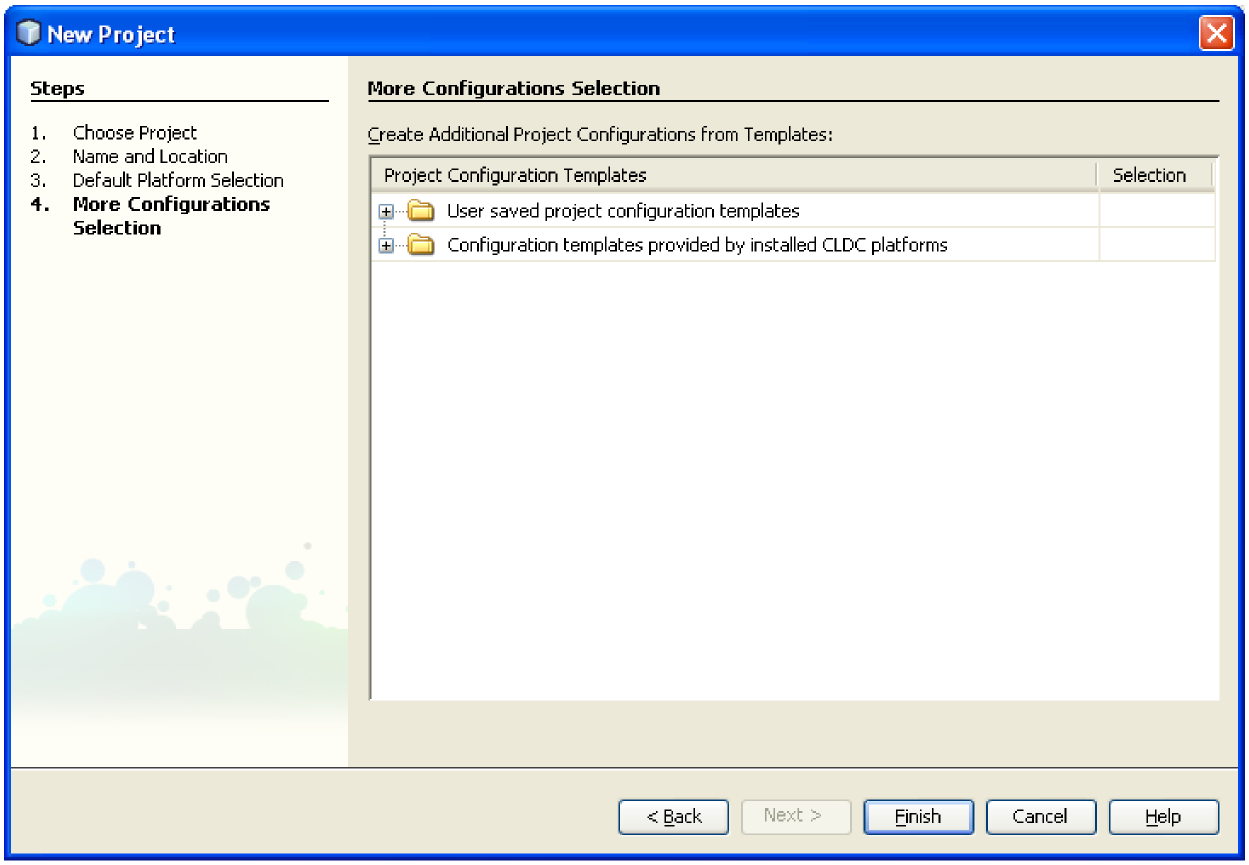
Emulator Platform: Sun Java(TM) Wireless Toolkit 2.5.2 for CLDC

Device: DefaultColorPhone

Device Configuration: CLDC-1.0 [✓ CLDC-1.1]

Device Profile: MIDP-1.0 [✓ MIDP-2.0] MIDP-2.1

5. Click Finish.


New Project

Steps

1. Choose Project
2. Name and Location
3. Default Platform Selection
4. **More Configurations Selection**

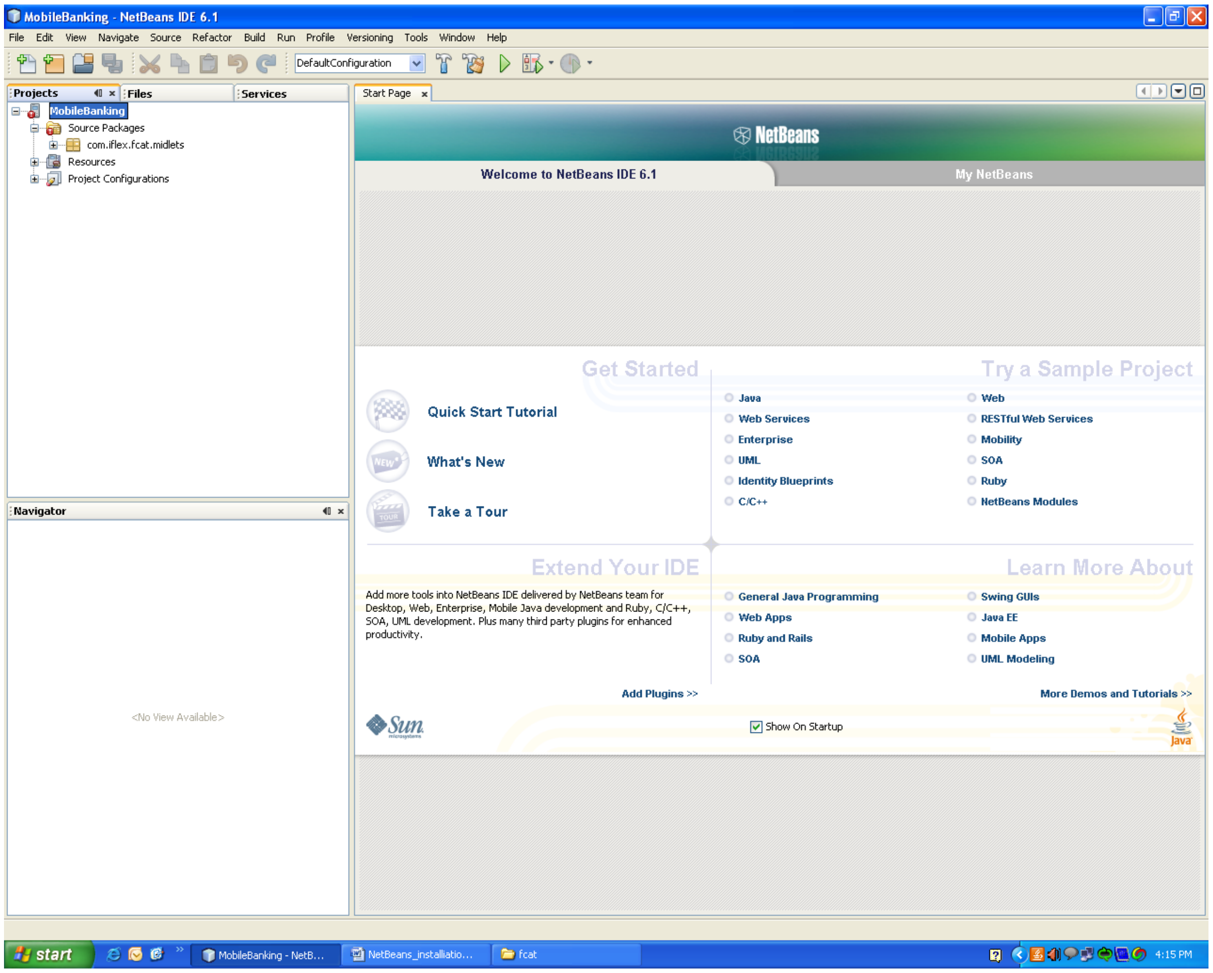
More Configurations Selection

Create Additional Project Configurations from Templates:

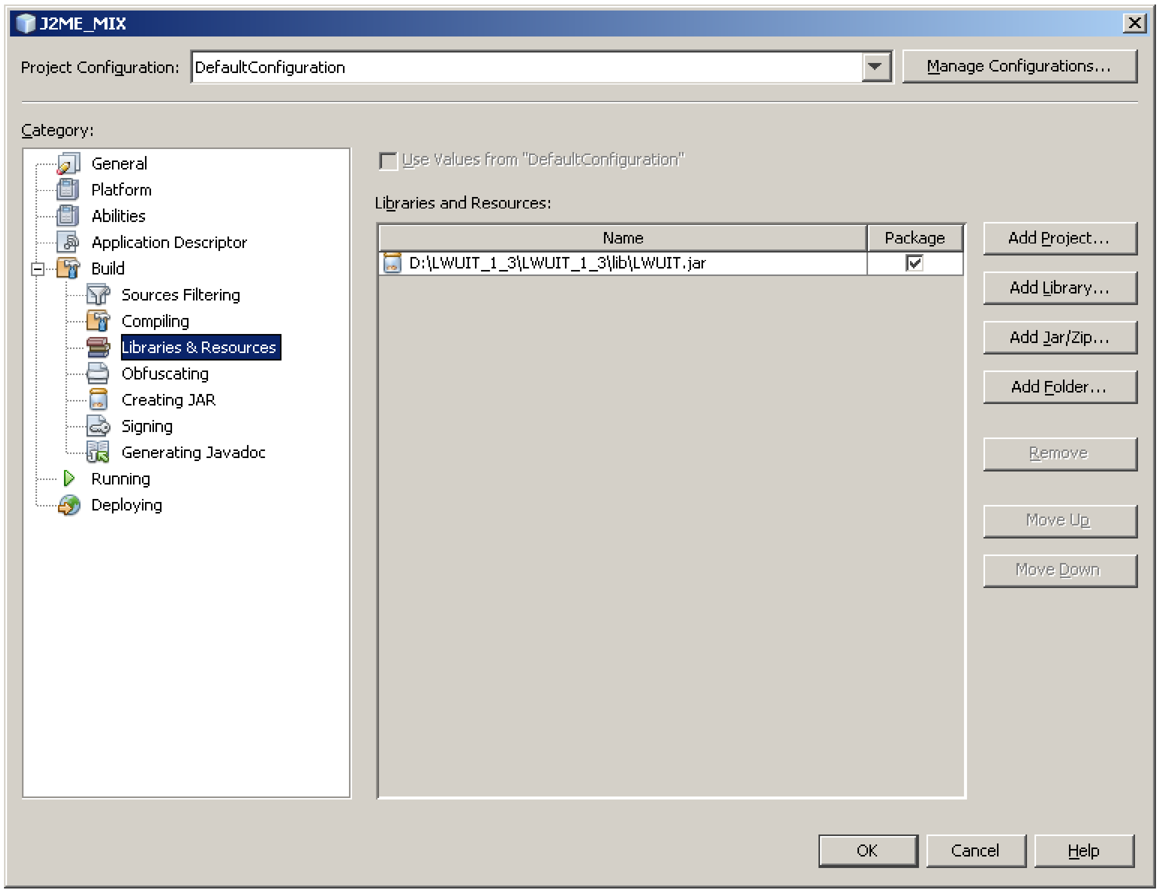
Project Configuration Templates	Selection
User saved project configuration templates	
Configuration templates provided by installed CLDC platforms	

6.After successful creation of Project, 4 folders namely 'build',' dist',' nbproject','src' will be created inside the base folder mentioned in step 3.

Copy the com/iflex/fcat/midlets folder to src/com/iflex/fcat/midlets.
Copy images and resource file to src and refresh the project.



7. Go to Project Properties -> Libraries & Resources. Click on Add Jar/Zip and add LWUIT.1.3.jar. Click Ok.Also Add jp.jar ,zxing-core-1.6.jar , lcrypto-j2me-149.jar added as a part for J2ME 12.0.2 development for QR code support and RSA Encryption support.



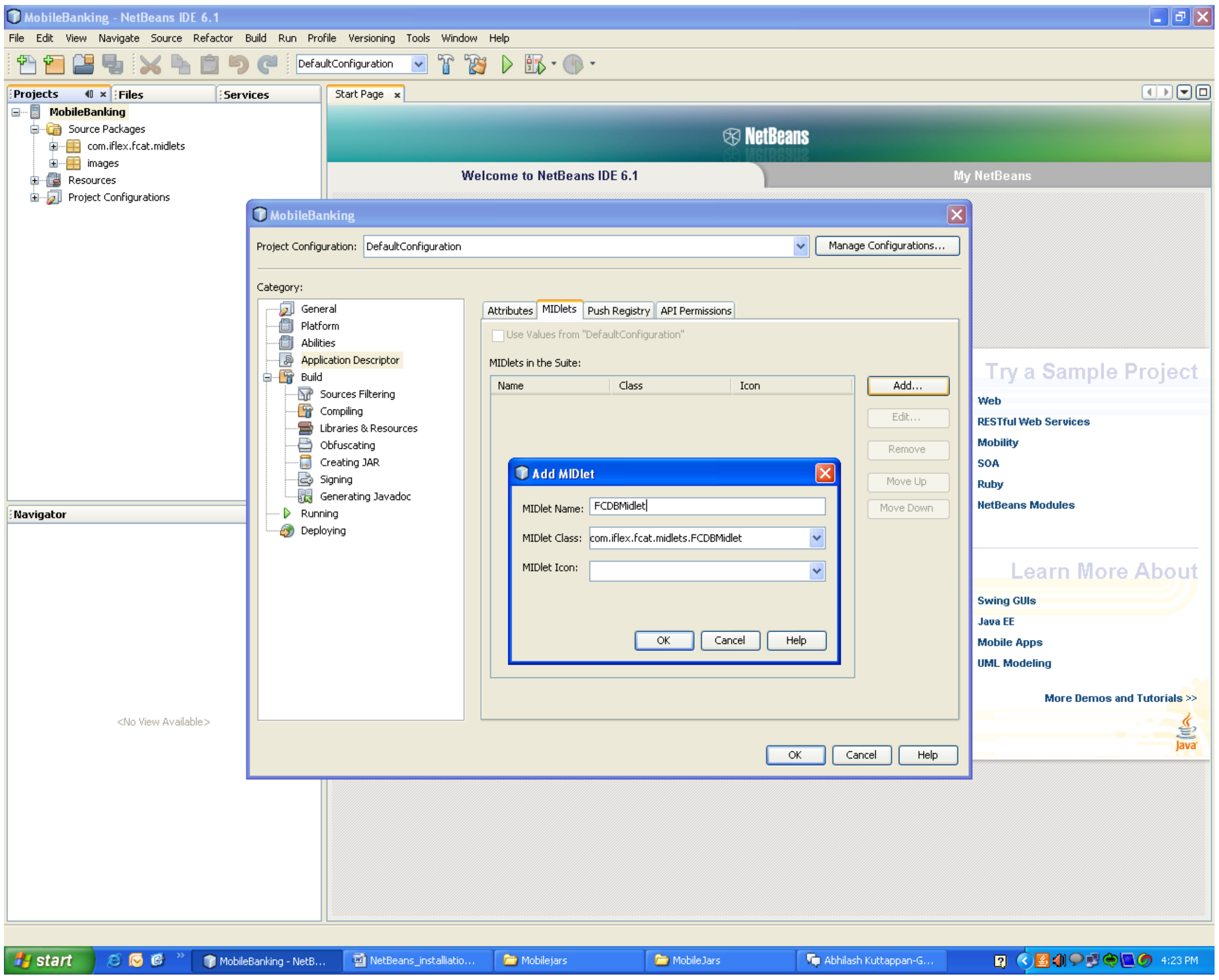
Project Configuration: DefaultConfiguration

Use Values from "DefaultConfiguration"

Libraries and Resources:

Name	Package
D:\LWUIT_1_3\LWUIT_1_3\lib\LWUIT.jar	✓

8. Go to Project Properties -> Application Descriptor -> Select Midlet Tab and click on Add. Select ‘FCDBMidlet’ from dropdown. Click Ok.

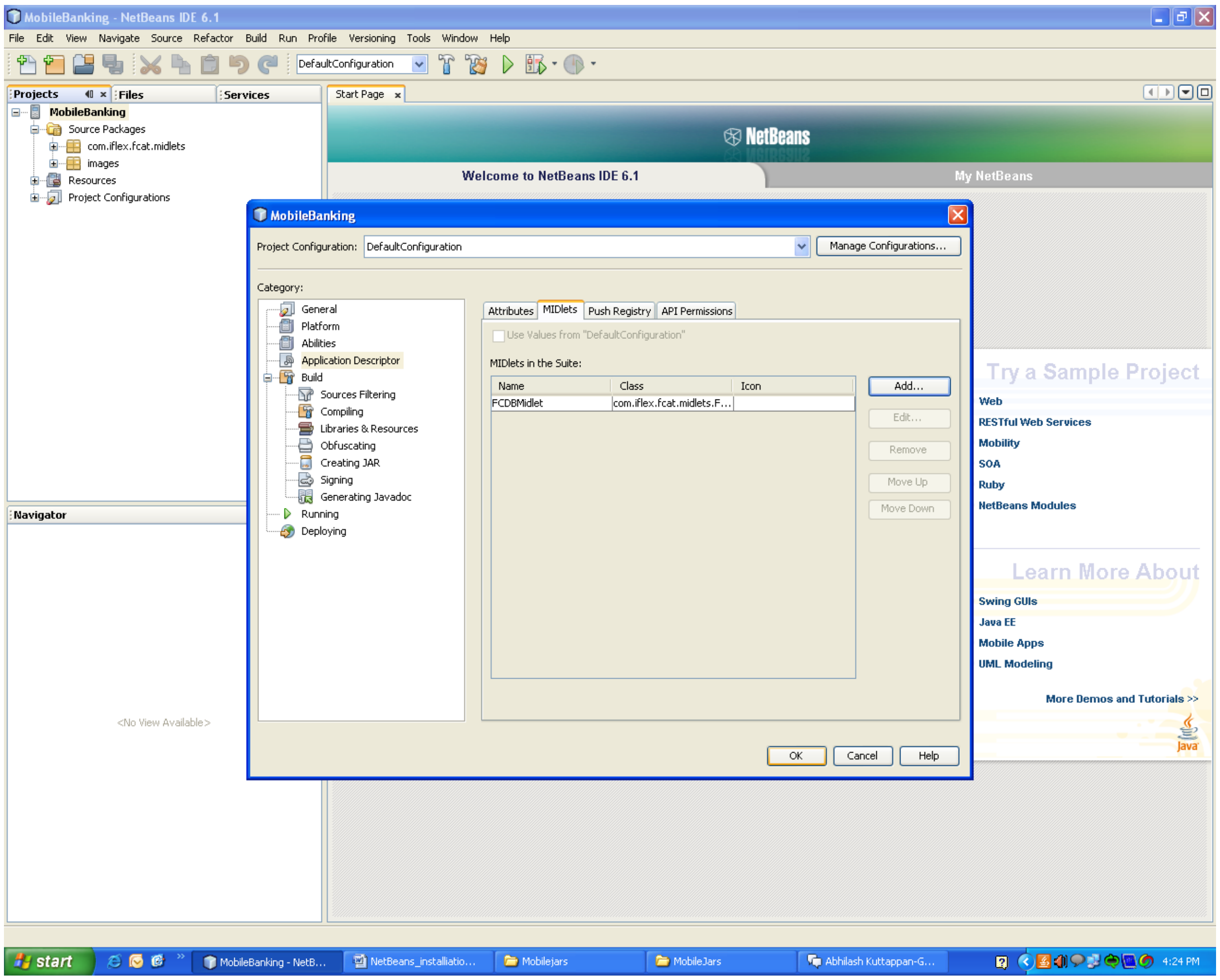


MIDlet Name: FCDBMidlet

MIDlet Class: com.iflex.fcat.midlets.FCDBMidlet

MIDlet Icon:

9.FCDBMidlet will be marked as a default Midlet.If you want to run Rich Midlet Use FCDBRichMidlet Click Ok.



MobileBanking

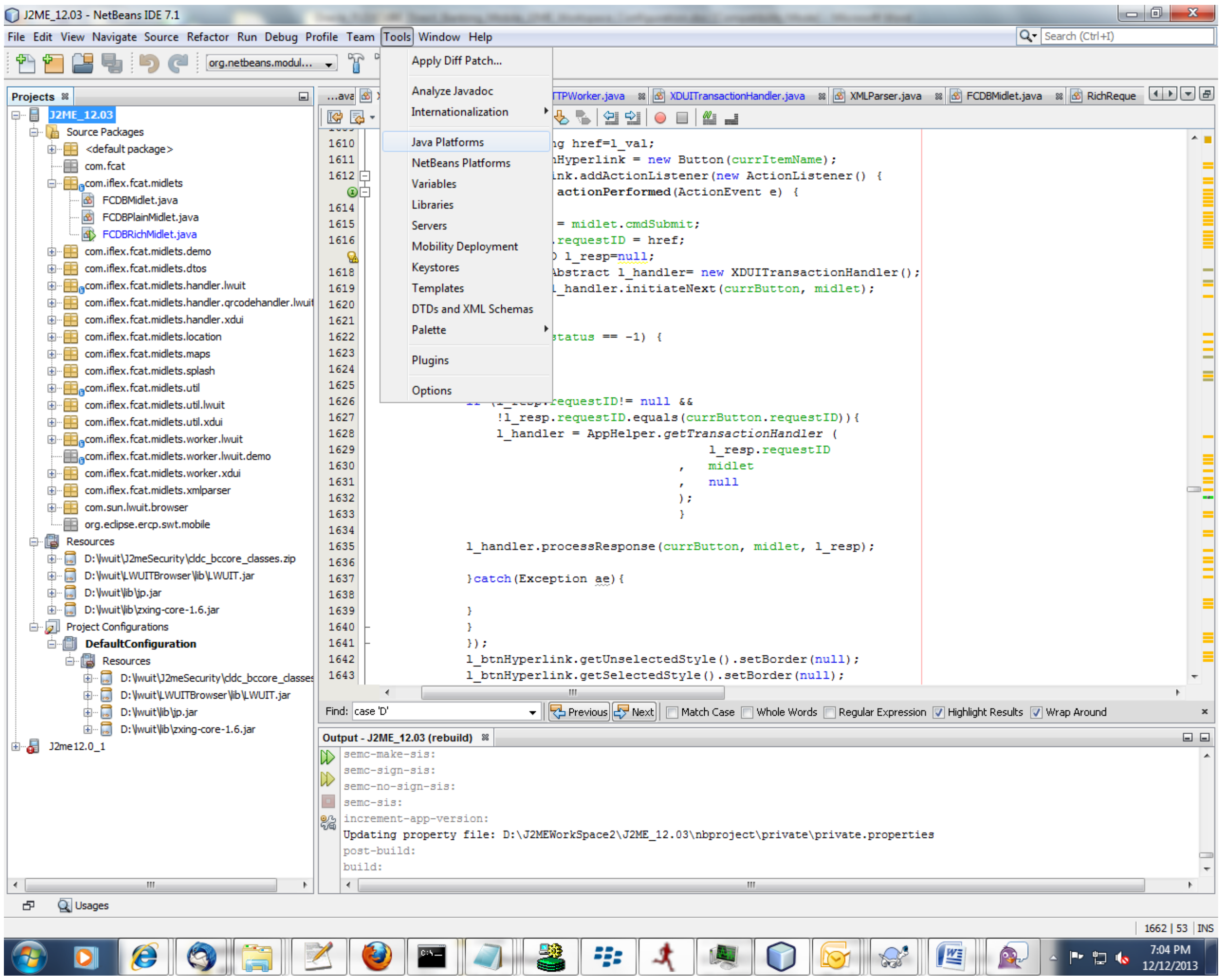
Project Configuration: DefaultConfiguration

Use Values from "DefaultConfiguration"

MIDlets in the Suite:

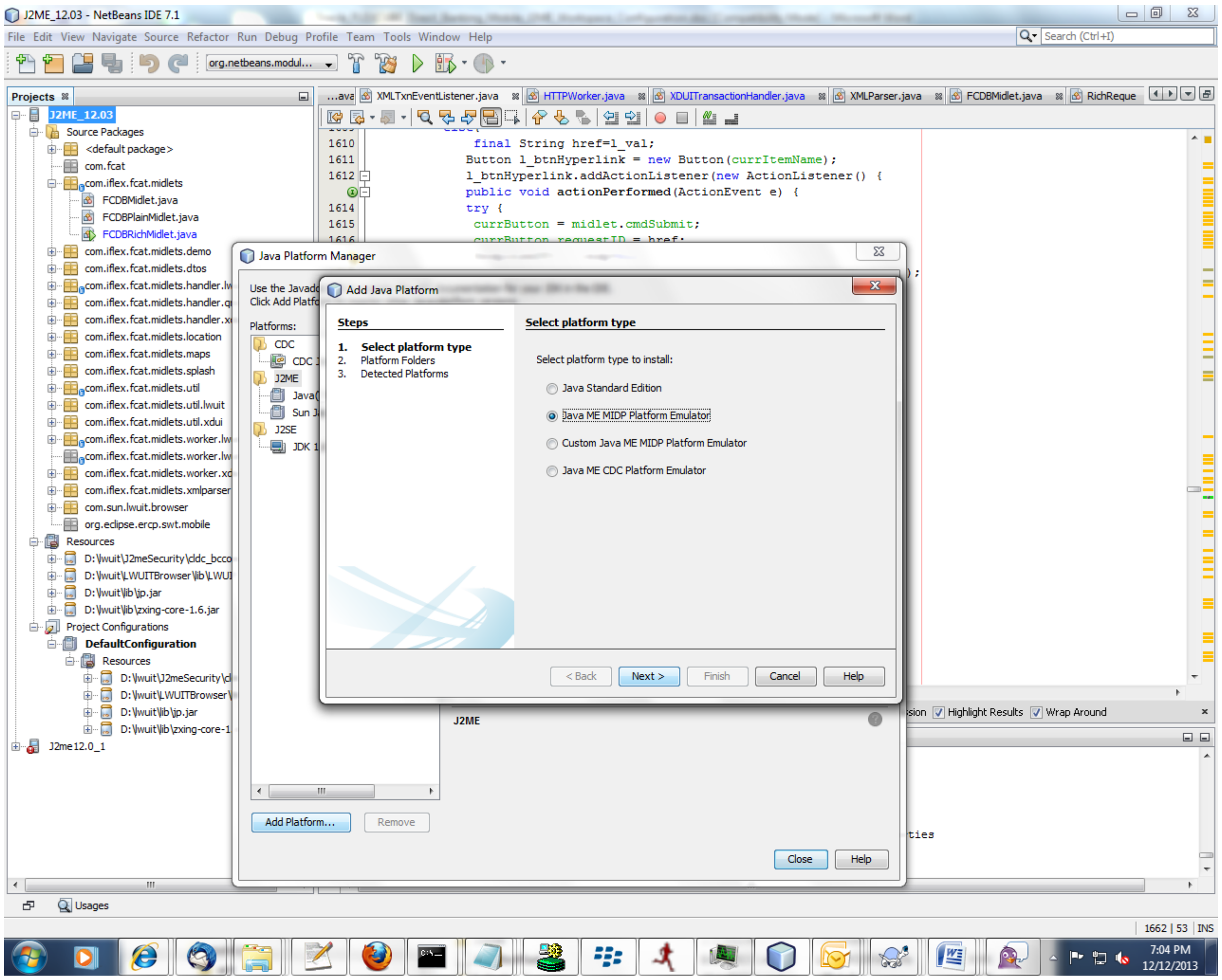
Name	Class	Icon
FCDBMidlet	com.iflex.fcat.midlets.F...	

10. Please add the platform for WTK 2.5.2 As it is being added in 12.0.2 Development. Steps are show below to add WTK env.


J2ME_12.03 - NetBeans IDE 7.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Apply Diff Patch...
Analyze Javadoc
Internationalization
Java Platforms
NetBeans Platforms
Variables
Libraries
Servers
Mobility Deployment
Keystores
Templates
DTDs and XML Schemas
Palette
Plugins
Options
Projects
J2ME_12.03
Source Packages
<default package>
com.fcat
com.iflex.fcat.midlets
FCDBMidlet.java
FCDBPlainMidlet.java
FCDBRichMidlet.java
com.iflex.fcat.midlets.demo
com.iflex.fcat.midlets.dtos
com.iflex.fcat.midlets.handler.lwuit
com.iflex.fcat.midlets.handler.qrcodehandler.lwuit
com.iflex.fcat.midlets.handler.xdui
com.iflex.fcat.midlets.location
com.iflex.fcat.midlets.maps
com.iflex.fcat.midlets.splash
com.iflex.fcat.midlets.util
com.iflex.fcat.midlets.util.lwuit
com.iflex.fcat.midlets.util.xdui
com.iflex.fcat.midlets.worker.lwuit
com.iflex.fcat.midlets.worker.lwuit.demo
com.iflex.fcat.midlets.worker.xdui
com.iflex.fcat.midlets.xmlparser
com.sun.lwuit.browser
org.eclipse.ercp.swt.mobile
Resources
D:\lwuit\J2meSecurity\cldc_bccore_classes.zip
D:\lwuit\LWUITBrowser\lib\LWUIT.jar
D:\lwuit\lib\jp.jar
D:\lwuit\lib\zxing-core-1.6.jar
Project Configurations
DefaultConfiguration
Resources
D:\lwuit\J2meSecurity\cldc_bccore_classes
D:\lwuit\LWUITBrowser\lib\LWUIT.jar
D:\lwuit\lib\jp.jar
D:\lwuit\lib\zxing-core-1.6.jar
J2me12.0_1
TPWorker.java
XDUITransactionHandler.java
XMLParser.java
FCDBMidlet.java
RichReque
Find: case 'D'
Previous
Next
Match Case
Whole Words
Regular Expression
Highlight Results
Wrap Around
Output - J2ME_12.03 (rebuild)
Usages
1662 | 53 INS
7:04 PM
12/12/2013

```
ng href=l_val;
nHyperlink = new Button(currItemName);
ink.addActionListener(new ActionListener() {
 actionPerformed(ActionEvent e) {

 = midlet.cmdSubmit;
.requestID = href;
) l_resp=null;
Abstract l_handler= new XDUITransactionHandler();
l_handler.initiateNext(currButton, midlet);


status == -1) {



    requestID!= null &&
     !l_resp.requestID.equals(currButton.requestID)){
     l_handler = AppHelper.getTransactionHandler (
                                  l_resp.requestID
                              ,   midlet
                              ,   null
                              );
                              }

l_handler.processResponse(currButton, midlet, l_resp);

}catch(Exception ae){

}
}
});
l_btnHyperlink.getUnselectedStyle().setBorder(null);
l_btnHyperlink.getSelectedStyle().setBorder(null);
```

```
semc-make-sis:
semc-sign-sis:
semc-no-sign-sis:
semc-sis:
increment-app-version:
Updating property file: D:\J2MEWorkSpace2\J2ME_12.03\nbproject\private\private.properties
post-build:
build:
```



```
        final String href=l_val;
       Button l_btnHyperlink = new Button(currItemName);
       l_btnHyperlink.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
       try {
        currButton = midlet.cmdSubmit;
```

Java Platform Manager

Add Java Platform

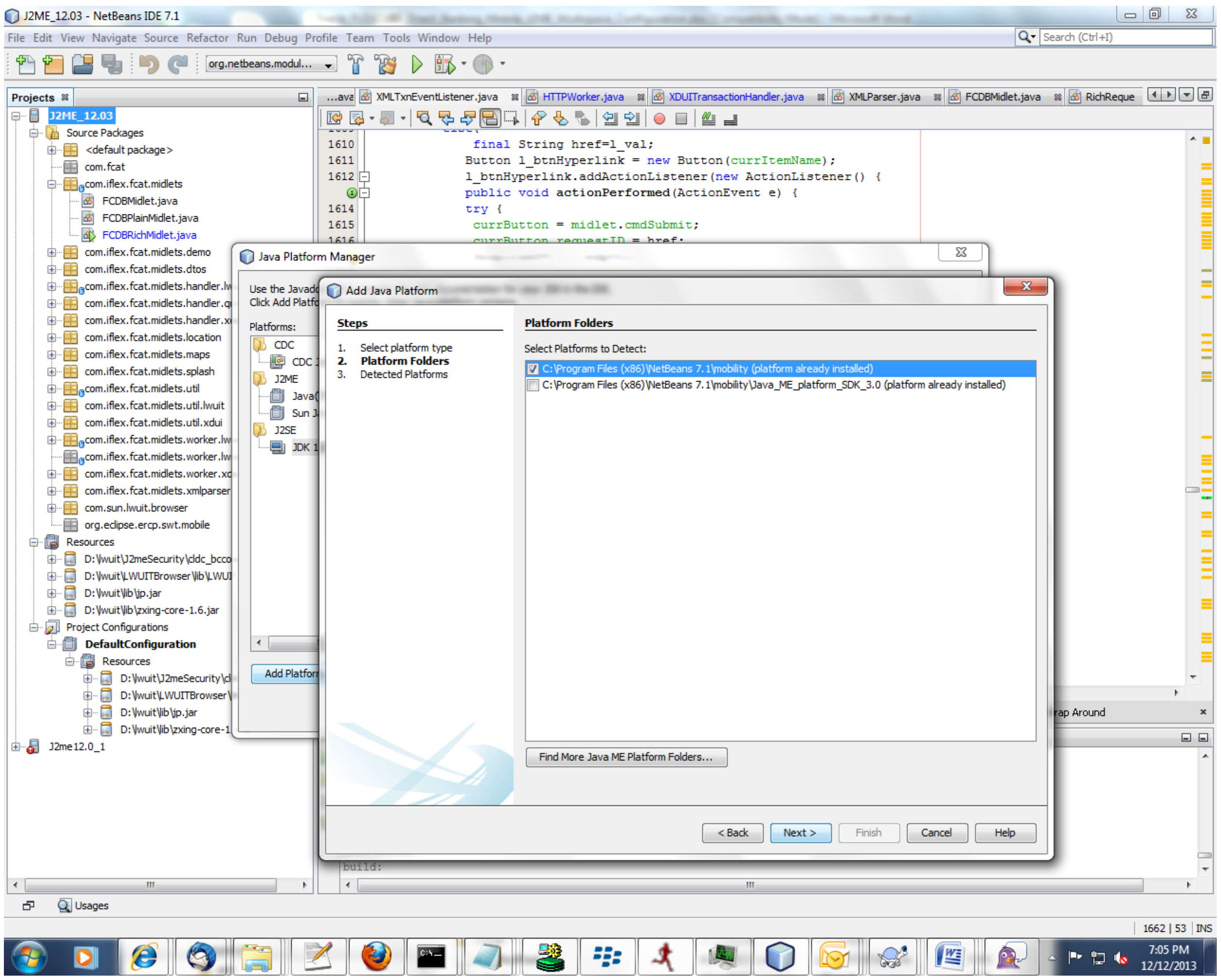
Steps

1. **Select platform type**
2. Platform Folders
3. Detected Platforms

Select platform type

Select platform type to install:

- ○ Java Standard Edition
- ● Java ME MIDP Platform Emulator
- ○ Custom Java ME MIDP Platform Emulator
- ○ Java ME CDC Platform Emulator


J2ME_12.03 - NetBeans IDE 7.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
Java Platform Manager
Add Java Platform
Add Platform...
Find More Java ME Platform Folders...
< Back
Next >
Finish
Cancel
Help

```
final String href=l_val;
Button l_btnHyperlink = new Button(currItemName);
l_btnHyperlink.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
 currButton = midlet.cmdSubmit;
```

Steps

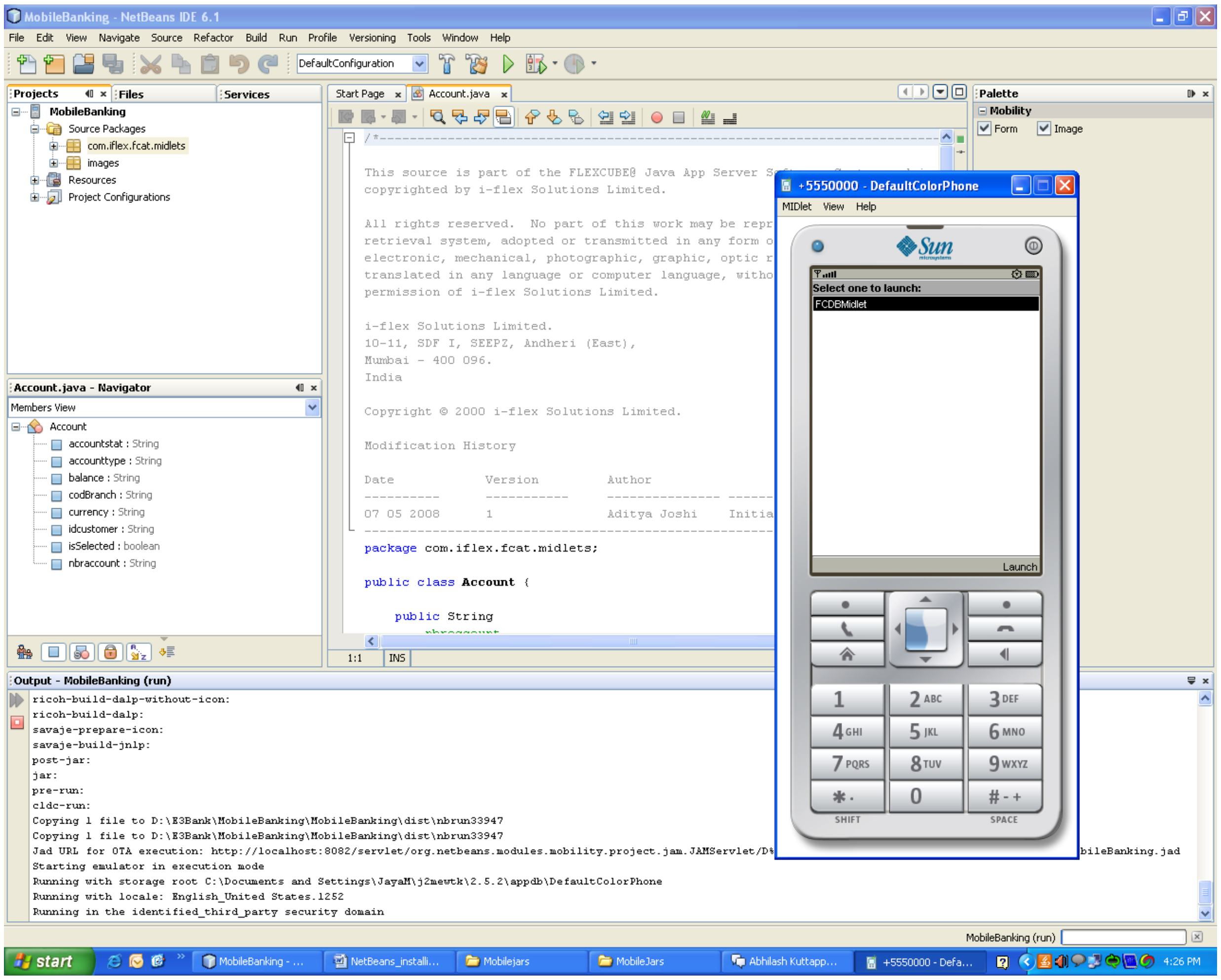
1. Select platform type
2. **Platform Folders**
3. Detected Platforms

Platform Folders

Select Platforms to Detect:

- [✓] C:\Program Files (x86)\NetBeans 7.1\mobility (platform already installed)
- [ ] C:\Program Files (x86)\NetBeans 7.1\mobility\Java_ME_platform_SDK_3.0 (platform already installed)

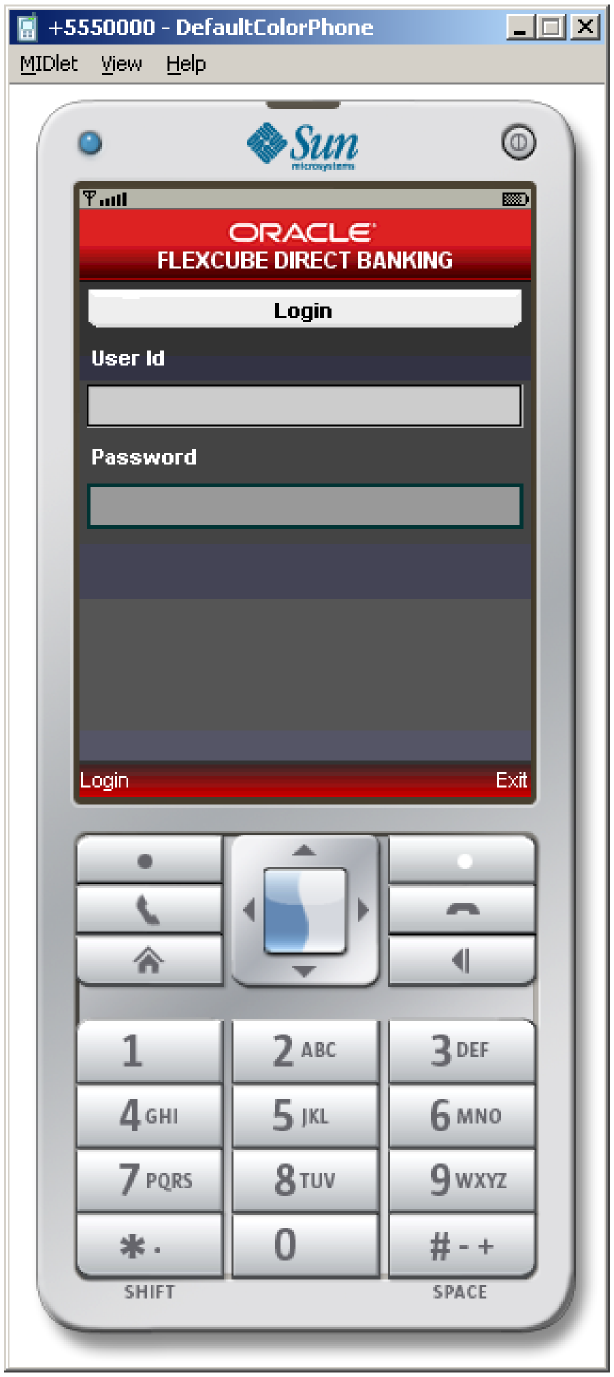
11. To run the application click on Run (Green Arrow). If configuration is proper, then following simulator will be shown.



```
/*-----------------------------------------------------------------------

This source is part of the FLEXCUBE@ Java App Server S
copyrighted by i-flex Solutions Limited.

All rights reserved.  No part of this work may be repr
retrieval system, adopted or transmitted in any form o
electronic, mechanical, photographic, graphic, optic r
translated in any language or computer language, witho
permission of i-flex Solutions Limited.

i-flex Solutions Limited.
10-11, SDF I, SEEPZ, Andheri (East),
Mumbai - 400 096.
India

Copyright © 2000 i-flex Solutions Limited.

Modification History

Date          Version        Author
----------    -----------    --------------- ------
07 05 2008    1              Aditya Joshi    Initia
-----------------------------------------------------
package com.iflex.fcat.midlets;

public class Account {

    public String
```

```
ricoh-build-dalp-without-icon:
ricoh-build-dalp:
savaje-prepare-icon:
savaje-build-jnlp:
post-jar:
jar:
pre-run:
cldc-run:
Copying 1 file to D:\E3Bank\MobileBanking\MobileBanking\dist\nbrun33947
Copying 1 file to D:\E3Bank\MobileBanking\MobileBanking\dist\nbrun33947
Jad URL for OTA execution: http://localhost:8082/servlet/org.netbeans.modules.mobility.project.jam.JAMServlet/D%                    bileBanking.jad
Starting emulator in execution mode
Running with storage root C:\Documents and Settings\JayaM\j2mewtk\2.5.2\appdb\DefaultColorPhone
Running with locale: English_United States.1252
Running in the identified_third_party security domain
```

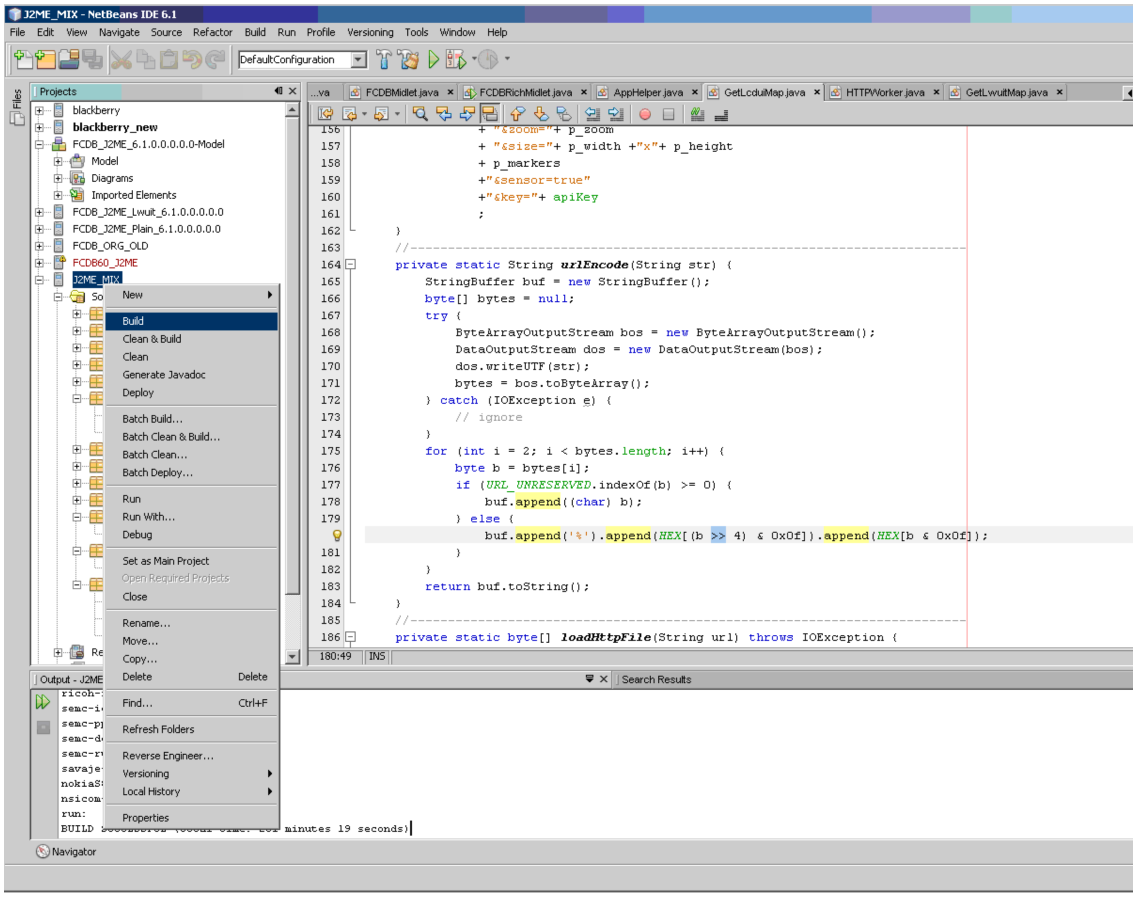
11. Click on ‘Launch’ button on the simulator to get the login screen.



2.1.2 Creating jar and jad

Using Build option user can create JAR/JAD

Right click on project select Option Build.



```
            + "&zoom="+ p_zoom
            + "&size="+ p_width +"x"+ p_height
            + p_markers
            +"&sensor=true"
            +"&key="+ apiKey
            ;
    }
    //-----------------------------------------------------------------------
    private static String urlEncode(String str) {
        StringBuffer buf = new StringBuffer();
        byte[] bytes = null;
        try {
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            DataOutputStream dos = new DataOutputStream(bos);
            dos.writeUTF(str);
            bytes = bos.toByteArray();
        } catch (IOException e) {
            // ignore
        }
        for (int i = 2; i < bytes.length; i++) {
            byte b = bytes[i];
            if (URL_UNRESERVED.indexOf(b) >= 0) {
                buf.append((char) b);
            } else {
                buf.append('%').append(HEX[(b >> 4) & 0x0f]).append(HEX[b & 0x0f]);
            }
        }
        return buf.toString();
    }
    //-----------------------------------------------------------------------
    private static byte[] loadHttpFile(String url) throws IOException {
```

```
ricoh-
semc-i
semc-p
semc-d
semc-r
savaje
nokiaS
nsicom
run:
BUILD                              minutes 19 seconds)
```

JAR/JAD will be created inside folder dist

Projectfolder > dist

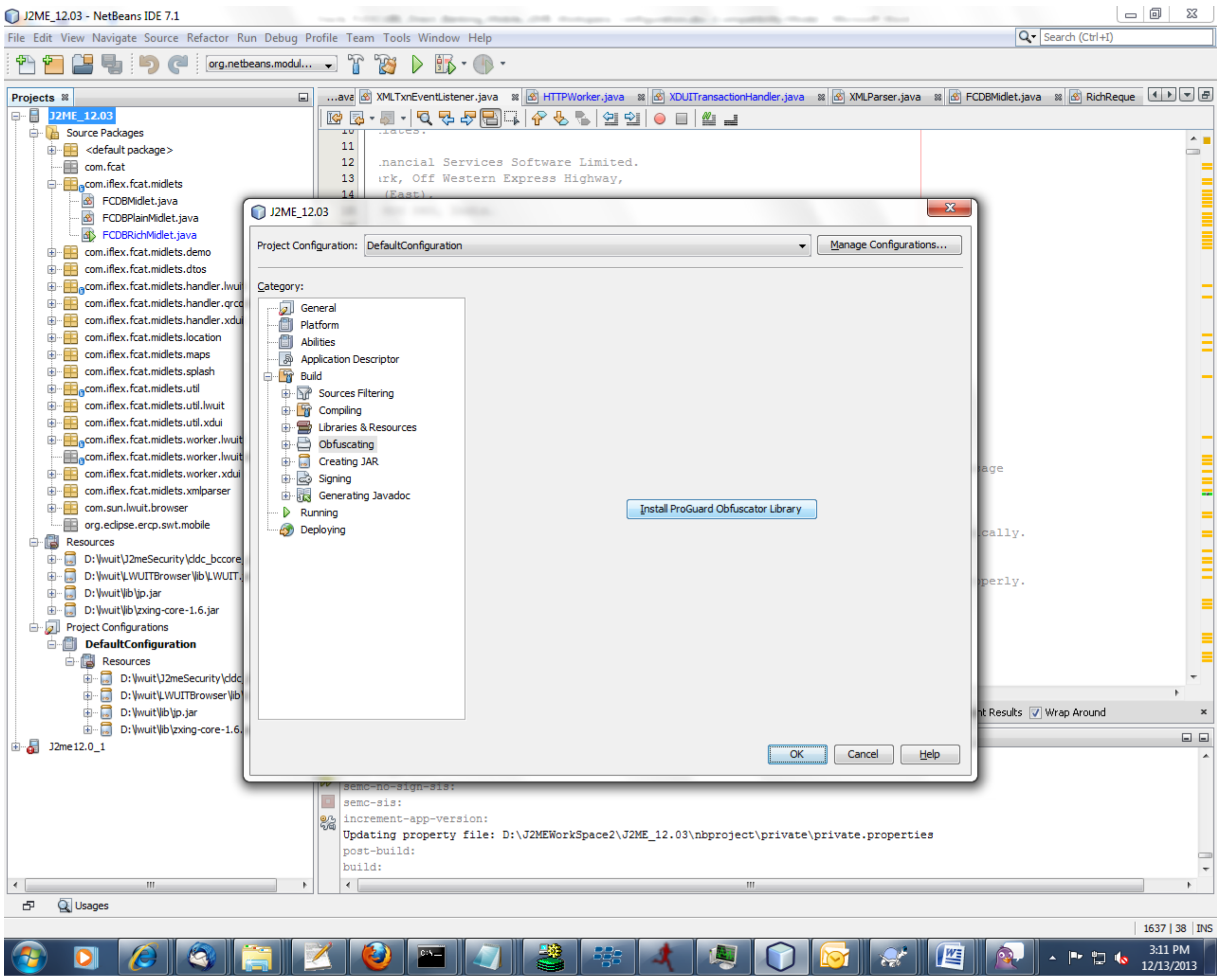
3. Obfuscation

3.1 Introduction

Code obfuscation is the process of "obscuring" Java classes with the purpose of making them harder to decompile back into source (reverse-engineer). Obfuscation typically entails renaming classes, methods and fields to use shorter names, effectively reducing the size of Java class.

3.2 Adding Obfuscating Parameters for New WTK 2.5.2

1.Right Click on J2ME Project name->Properties->Obfuscating



Project Configuration: DefaultConfiguration

```
semc-no-sign-sis:
semc-sis:
increment-app-version:
Updating property file: D:\J2MEWorkSpace2\J2ME_12.03\nbproject\private\private.properties
post-build:
build:
```

1. Click on Install Obfuscator Proguard Library
2. Select Obfuscation Level High And put the following parameters under "Additional Obfuscation Settings"

-optimizationpasses 3

-printmapping 'D:\FCDB\system\build\make\apache-ant-1.8.0\FCDB\mobile\proguard\obfuscate\j2merich.log'

-microedition

-dontusemixedcaseclassnames

*-keep public class com.iflex.fcat.midlets.handler.lwuit.**

# Keep - Applications. Keep all application classes, along with their 'main'

```
# methods.

-keepclasseswithmembers public class * {

    public static void main(java.lang.String[]);

}


# Keep - Midlets. Keep all extensions of javax.microedition.midlet.MIDlet.

-keep public class com.iflex.fcat.midlets.FCDBRichMidlet extends javax.microedition.midlet.MIDlet


# Also keep - Enumerations. Keep the special static methods that are required in

# enumeration classes.

-keepclassmembers enum  * {

    public static **[] values();

    public static ** valueOf(java.lang.String);

}


# Also keep - Database drivers. Keep all implementations of java.sql.Driver.

-keep class * extends java.sql.Driver


# Also keep - Swing UI L&F. Keep all extensions of javax.swing.plaf.ComponentUI,

# along with the special 'createUI' method.

-keep class * extends javax.swing.plaf.ComponentUI {

    public static javax.swing.plaf.ComponentUI createUI(javax.swing.JComponent);

}


# Keep names - Native method names. Keep all native class/method names.

-keepclasseswithmembers,allowshrinking class * {

    native <methods>;

}
```

```
# Remove - System method calls. Remove all invocations of System
# methods without side effects whose return values are not used.
-assumenosideeffects public class java.lang.System {
    public static long currentTimeMillis();
    static java.lang.Class getCallerClass();
    public static int identityHashCode(java.lang.Object);
    public static java.lang.SecurityManager getSecurityManager();
    public static java.util.Properties getProperties();
    public static java.lang.String getProperty(java.lang.String);
    public static java.lang.String getenv(java.lang.String);
    public static java.lang.String mapLibraryName(java.lang.String);
    public static java.lang.String getProperty(java.lang.String,java.lang.String);
}

# Remove - Math method calls. Remove all invocations of Math
# methods without side effects whose return values are not used.
-assumenosideeffects public class java.lang.Math {
    public static double sin(double);
    public static double cos(double);
    public static double tan(double);
    public static double asin(double);
    public static double acos(double);
    public static double atan(double);
    public static double toRadians(double);
    public static double toDegrees(double);
    public static double exp(double);
    public static double log(double);
    public static double log10(double);
    public static double sqrt(double);
```

public static double cbrt(double);

public static double IEEEremainder(double,double);

public static double ceil(double);

public static double floor(double);

public static double rint(double);

public static double atan2(double,double);

public static double pow(double,double);

public static int round(float);

public static long round(double);

public static double random();

public static int abs(int);

public static long abs(long);

public static float abs(float);

public static double abs(double);

public static int max(int,int);

public static long max(long,long);

public static float max(float,float);

public static double max(double,double);

public static int min(int,int);

public static long min(long,long);

public static float min(float,float);

public static double min(double,double);

public static double ulp(double);

public static float ulp(float);

public static double signum(double);

public static float signum(float);

public static double sinh(double);

public static double cosh(double);

public static double tanh(double);

```
    public static double hypot(double,double);

    public static double expm1(double);

    public static double log1p(double);

}


# Remove - Number method calls. Remove all invocations of Number

# methods without side effects whose return values are not used.

-assumenosideeffects public class java.lang.* extends java.lang.Number {

    public static java.lang.String toString(byte);

    public static java.lang.Byte valueOf(byte);

    public static byte parseByte(java.lang.String);

    public static byte parseByte(java.lang.String,int);

    public static java.lang.Byte valueOf(java.lang.String,int);

    public static java.lang.Byte valueOf(java.lang.String);

    public static java.lang.Byte decode(java.lang.String);

    public int compareTo(java.lang.Byte);

    public static java.lang.String toString(short);

    public static short parseShort(java.lang.String);

    public static short parseShort(java.lang.String,int);

    public static java.lang.Short valueOf(java.lang.String,int);

    public static java.lang.Short valueOf(java.lang.String);

    public static java.lang.Short valueOf(short);

    public static java.lang.Short decode(java.lang.String);

    public static short reverseBytes(short);

    public int compareTo(java.lang.Short);

    public static java.lang.String toString(int,int);

    public static java.lang.String toHexString(int);

    public static java.lang.String toOctalString(int);

    public static java.lang.String toBinaryString(int);
```

public static java.lang.String toString(int);

public static int parseInt(java.lang.String,int);

public static int parseInt(java.lang.String);

public static java.lang.Integer valueOf(java.lang.String,int);

public static java.lang.Integer valueOf(java.lang.String);

public static java.lang.Integer valueOf(int);

public static java.lang.Integer getInteger(java.lang.String);

public static java.lang.Integer getInteger(java.lang.String,int);

public static java.lang.Integer getInteger(java.lang.String,java.lang.Integer);

public static java.lang.Integer decode(java.lang.String);

public static int highestOneBit(int);

public static int lowestOneBit(int);

public static int numberOfLeadingZeros(int);

public static int numberOfTrailingZeros(int);

public static int bitCount(int);

public static int rotateLeft(int,int);

public static int rotateRight(int,int);

public static int reverse(int);

public static int signum(int);

public static int reverseBytes(int);

public int compareTo(java.lang.Integer);

public static java.lang.String toString(long,int);

public static java.lang.String toHexString(long);

public static java.lang.String toOctalString(long);

public static java.lang.String toBinaryString(long);

public static java.lang.String toString(long);

public static long parseLong(java.lang.String,int);

public static long parseLong(java.lang.String);

public static java.lang.Long valueOf(java.lang.String,int);

public static java.lang.Long valueOf(java.lang.String);

public static java.lang.Long valueOf(long);

public static java.lang.Long decode(java.lang.String);

public static java.lang.Long getLong(java.lang.String);

public static java.lang.Long getLong(java.lang.String,long);

public static java.lang.Long getLong(java.lang.String,java.lang.Long);

public static long highestOneBit(long);

public static long lowestOneBit(long);

public static int numberOfLeadingZeros(long);

public static int numberOfTrailingZeros(long);

public static int bitCount(long);

public static long rotateLeft(long,int);

public static long rotateRight(long,int);

public static long reverse(long);

public static int signum(long);

public static long reverseBytes(long);

public int compareTo(java.lang.Long);

public static java.lang.String toString(float);

public static java.lang.String toHexString(float);

public static java.lang.Float valueOf(java.lang.String);

public static java.lang.Float valueOf(float);

public static float parseFloat(java.lang.String);

public static boolean isNaN(float);

public static boolean isInfinite(float);

public static int floatToIntBits(float);

public static int floatToRawIntBits(float);

public static float intBitsToFloat(int);

public static int compare(float,float);

public boolean isNaN();

public boolean isInfinite();

public int compareTo(java.lang.Float);

public static java.lang.String toString(double);

public static java.lang.String toHexString(double);

public static java.lang.Double valueOf(java.lang.String);

public static java.lang.Double valueOf(double);

public static double parseDouble(java.lang.String);

public static boolean isNaN(double);

public static boolean isInfinite(double);

public static long doubleToLongBits(double);

public static long doubleToRawLongBits(double);

public static double longBitsToDouble(long);

public static int compare(double,double);

public boolean isNaN();

public boolean isInfinite();

public int compareTo(java.lang.Double);

public <init>(byte);

public <init>(short);

public <init>(int);

public <init>(long);

public <init>(float);

public <init>(double);

public <init>(java.lang.String);

public byte byteValue();

public short shortValue();

public int intValue();

public long longValue();

public float floatValue();

public double doubleValue();

```
    public int compareTo(java.lang.Object);

    public boolean equals(java.lang.Object);

    public int hashCode();

    public java.lang.String toString();

}


# Remove - String method calls. Remove all invocations of String

# methods without side effects whose return values are not used.

-assumenosideeffects public class java.lang.String {

    public <init>();

    public <init>(byte[]);

    public <init>(byte[],int);

    public <init>(byte[],int,int);

    public <init>(byte[],int,int,int);

    public <init>(byte[],int,int,java.lang.String);

    public <init>(byte[],java.lang.String);

    public <init>(char[]);

    public <init>(char[],int,int);

    public <init>(java.lang.String);

    public <init>(java.lang.StringBuffer);

    public static java.lang.String copyValueOf(char[]);

    public static java.lang.String copyValueOf(char[],int,int);

    public static java.lang.String valueOf(boolean);

    public static java.lang.String valueOf(char);

    public static java.lang.String valueOf(char[]);

    public static java.lang.String valueOf(char[],int,int);

    public static java.lang.String valueOf(double);

    public static java.lang.String valueOf(float);

    public static java.lang.String valueOf(int);
```

public static java.lang.String valueOf(java.lang.Object);

public static java.lang.String valueOf(long);

public boolean contentEquals(java.lang.StringBuffer);

public boolean endsWith(java.lang.String);

public boolean equalsIgnoreCase(java.lang.String);

public boolean equals(java.lang.Object);

public boolean matches(java.lang.String);

public boolean regionMatches(boolean,int,java.lang.String,int,int);

public boolean regionMatches(int,java.lang.String,int,int);

public boolean startsWith(java.lang.String);

public boolean startsWith(java.lang.String,int);

public byte[] getBytes();

public byte[] getBytes(java.lang.String);

public char charAt(int);

public char[] toCharArray();

public int compareToIgnoreCase(java.lang.String);

public int compareTo(java.lang.Object);

public int compareTo(java.lang.String);

public int hashCode();

public int indexOf(int);

public int indexOf(int,int);

public int indexOf(java.lang.String);

public int indexOf(java.lang.String,int);

public int lastIndexOf(int);

public int lastIndexOf(int,int);

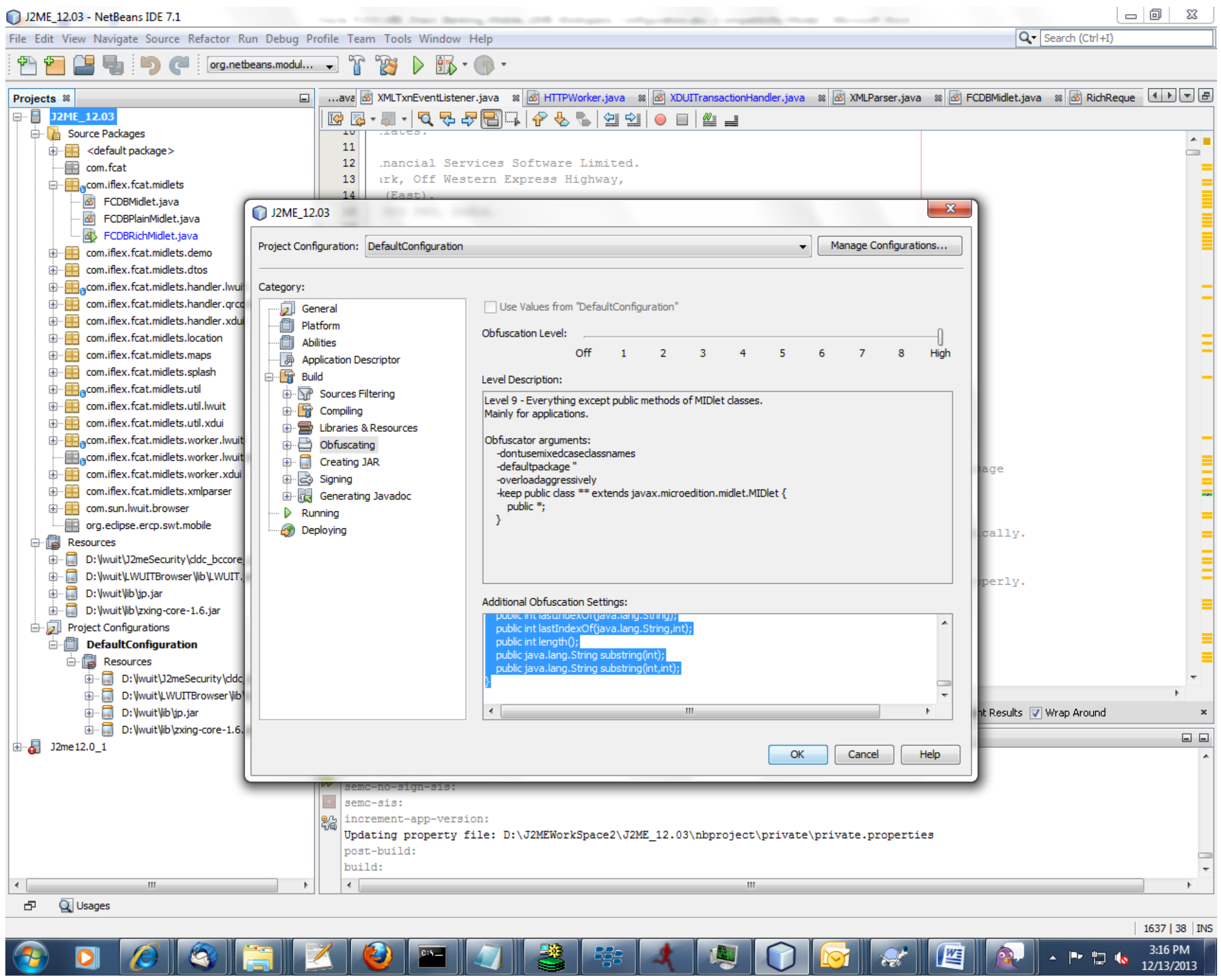
public int lastIndexOf(java.lang.String);

public int lastIndexOf(java.lang.String,int);

public int length();

public java.lang.CharSequence subSequence(int,int);

```
    public java.lang.String concat(java.lang.String);

    public java.lang.String replaceAll(java.lang.String,java.lang.String);

    public java.lang.String replace(char,char);

    public java.lang.String replaceFirst(java.lang.String,java.lang.String);

    public java.lang.String[] split(java.lang.String);

    public java.lang.String[] split(java.lang.String,int);

    public java.lang.String substring(int);

    public java.lang.String substring(int,int);

    public java.lang.String toLowerCase();

    public java.lang.String toLowerCase(java.util.Locale);

    public java.lang.String toString();

    public java.lang.String toUpperCase();

    public java.lang.String toUpperCase(java.util.Locale);

    public java.lang.String trim();

}


# Remove - StringBuffer method calls. Remove all invocations of StringBuffer

# methods without side effects whose return values are not used.

-assumenosideeffects public class java.lang.StringBuffer {

    public <init>();

    public <init>(int);

    public <init>(java.lang.String);

    public <init>(java.lang.CharSequence);

    public java.lang.String toString();

    public char charAt(int);

    public int capacity();

    public int codePointAt(int);

    public int codePointBefore(int);

    public int indexOf(java.lang.String,int);
    public int lastIndexOf(java.lang.String);

    public int lastIndexOf(java.lang.String,int);

    public int length();

    public java.lang.String substring(int);

    public java.lang.String substring(int,int);

}


# Remove - StringBuilder method calls. Remove all invocations of StringBuilder

# methods without side effects whose return values are not used.

-assumenosideeffects public class java.lang.StringBuilder {

    public <init>();

    public <init>(int);

    public <init>(java.lang.String);

    public <init>(java.lang.CharSequence);

    public java.lang.String toString();

    public char charAt(int);

    public int capacity();

    public int codePointAt(int);

    public int codePointBefore(int);

    public int indexOf(java.lang.String,int);

    public int lastIndexOf(java.lang.String);

    public int lastIndexOf(java.lang.String,int);

    public int length();

    public java.lang.String substring(int);

    public java.lang.String substring(int,int);

}
```


J2ME_12.03 - NetBeans IDE 7.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects
J2ME_12.03
J2ME_12.03
OK
Cancel
Help
Manage Configurations...

Project Configuration: DefaultConfiguration

Category: General, Platform, Abilities, Application Descriptor, Build, Sources Filtering, Compiling, Libraries & Resources, Obfuscating, Creating JAR, Signing, Generating Javadoc, Running, Deploying

Use Values from "DefaultConfiguration"

Obfuscation Level: Off 1 2 3 4 5 6 7 8 High

Level Description:

Level 9 - Everything except public methods of MIDlet classes.
Mainly for applications.

Obfuscator arguments:
```
-dontusemixedcaseclassnames
-defaultpackage ''
-overloadaggressively
-keep public class ** extends javax.microedition.midlet.MIDlet {
  public *;
}
```

Additional Obfuscation Settings:
```
public int lastIndexOf(java.lang.String);
public int lastIndexOf(java.lang.String,int);
public int length();
public java.lang.String substring(int);
public java.lang.String substring(int,int);
}
```

```
semc-no-sign-sis:
semc-sis:
increment-app-version:
Updating property file: D:\J2MEWorkSpace2\J2ME_12.03\nbproject\private\private.properties
post-build:
build:
```

4. Click ok