

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### Table of Contents

REVISION HISTORY.....	2
COPYRIGHT.....	2
DISCLAIMER.....	2
INTRODUCTION.....	3
BENEFITS.....	3
REQUIREMENTS.....	3
LIMITATIONS.....	3
INSTALLATION.....	4
SETTING-UP ELIPSE TO USE SMS API.....	5
JAVA QUICK GUIDE.....	9
SENDING SMS.....	9
READING SMS.....	9
JAVA SAMPLE CODE.....	10
TERMS AND CONDITIONS ON THE USE OF THE JAVA SAMPLE CODE.....	15
TERMS AND CONDITIONS ON SUPPORT.....	15

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### REVISION HISTORY

<b><i>EDITION</i></b>	<b><i>ISSUED DATE</i></b>	<b><i>REMARK</i></b>
1 <sup>st</sup>	13 <sup>th</sup> of November, 2008	Draft release

### COPYRIGHT

Copyright © MOBITEK System Sdn. Bhd. 2008. All rights reserved. No part of this document may be reproduced, distributed, stored in a retrieval system or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, photocopying, manual or otherwise, without the prior written permission of MOBITEK System Sdn. Bhd.

### DISCLAIMER

MOBITEK makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, MOBITEK reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation to notify any person of such revision or changes. Trademarks and Registered Trademarks Products and product names mentioned in this document may be trademarks or registered trademarks of their respective owners.

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### INTRODUCTION

Our **SMS Gateway Development Kit** (SMS GDK) contains various kits or software components. **SMS GDK** allows system integrator and software developer (SI/SD) to build SMS gateway, SMS application and SMS solution.

One of the kits or components is **SMS Application Programming Interface** (SMS API ). The type of API is Microsoft's Component Object Model (COM) in the form of an **ActiveX Dll** file. We will call it **COM-ActiveX**.

**COM-ActiveX** is a Microsoft's technology (<http://www.microsoft.com/com/default.msp>). In order to use **COM-ActiveX** , SI/SD needs to use Microsoft's programming language or integrated development environment (IDE) such as VB6, VC++, Visual Studio, Visual Studio.Net, VB.Net and C#.

**Java** is Sun Microsystems' technology. Java cannot use **COM-ActiveX**.

A solution for Java developer is to use **Java COM Bridge** (JACOB). For more information, please refer to -- <http://danadler.com/jacob/>

With JACOB, SI/SD can use Java with our **SMS API**.

### BENEFITS

1. Use Java to build a SMS application.
2. SI/SD do not need to worry about the firmware changes to the GSM modem. Any underlying changes on GSM modem will not affect SI/SD's application or system.

### REQUIREMENTS

1. SI/SD must possess knowledge of Java programming, JRE and Eclipse (optional);
2. SI/SD must know how to use JACOB version 1.14.3;
3. JRE version 1.6 for Windows; and
4. Windows OS (except Vista).

### LIMITATIONS

1. Java application must run on Windows OS (except Vista). Linux not supported.
2. JACOB belongs to 3<sup>rd</sup> party.
3. We have not tested on JRE earlier than version 1.6.
4. Warranty of GSM Modem does not cover failure of Java to call SMS API via JACOB. The warranty only covers hardware failure.

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### INSTALLATION

1. Follow the instruction in CD to set-up the GSM modem.
2. Follow the instruction in CD to install **SMS API**.
3. Install JRE version 1.6 (if required). Available in the CD.
4. Install JACOB:
  - a) Unzip the file "jacob-1.14.3.zip" that is in CD
  - b) Copy these 2 files to "system32" folder:
    - jacob-1.14.3-x64.dll
    - jacob-1.14.3-x86.dll
5. In our example, we use Eclipse (available in the CD) as the IDE for Java.

Note:

*CD refers to the SMS Gateway Development Kit CD that comes with GSM modem Type SGDK.*

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

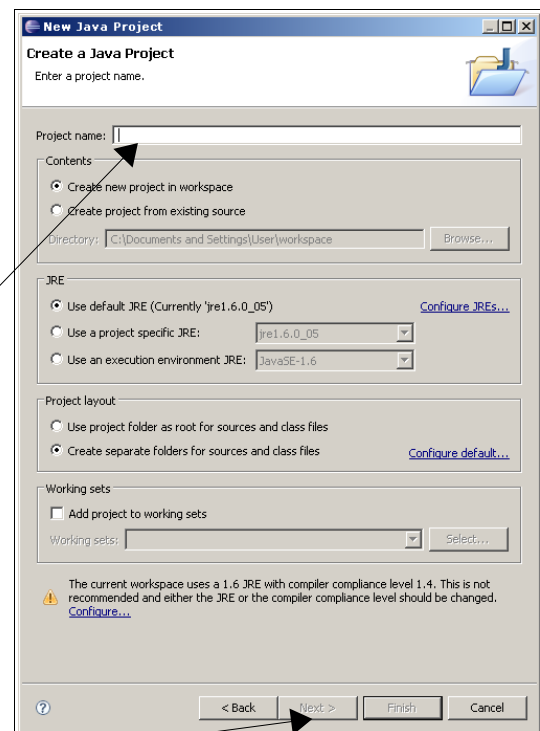
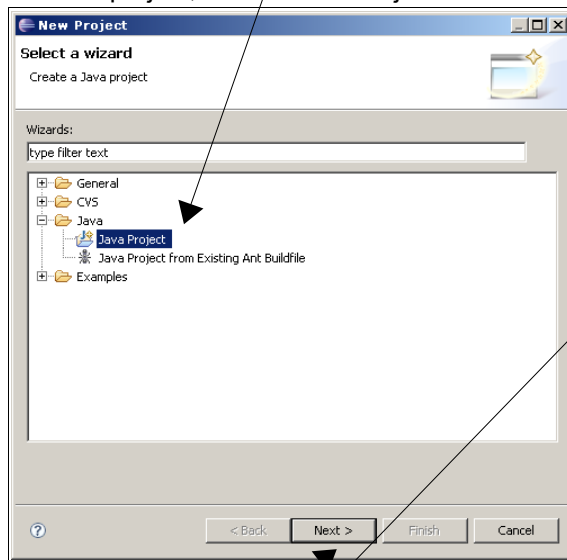
## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### SETTING-UP ELIPSE TO USE SMS API

Objective:-

- To show you how to set up Eclipse with JACOB
- To show show you how to use **SMS API** in Java language
- To use Java to send and read SMS

#### 1. Add a new project, select “Java Project”



#### 2. Click “Next”

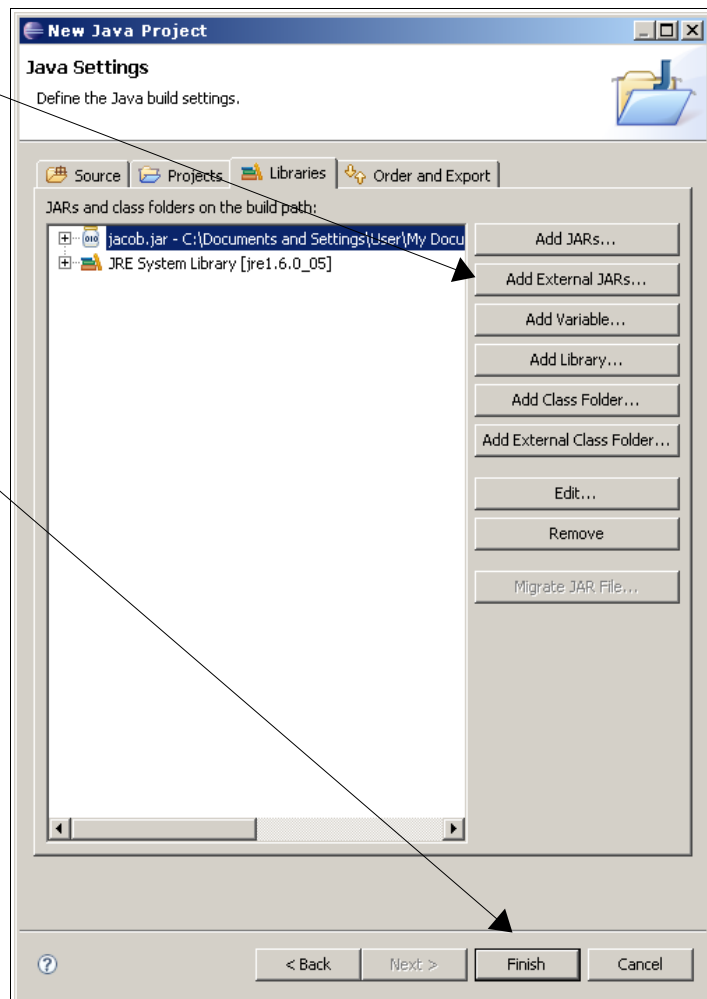
#### 3. Type “JavaSMS” in “Project Name:”

#### 4. Click “Next”

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

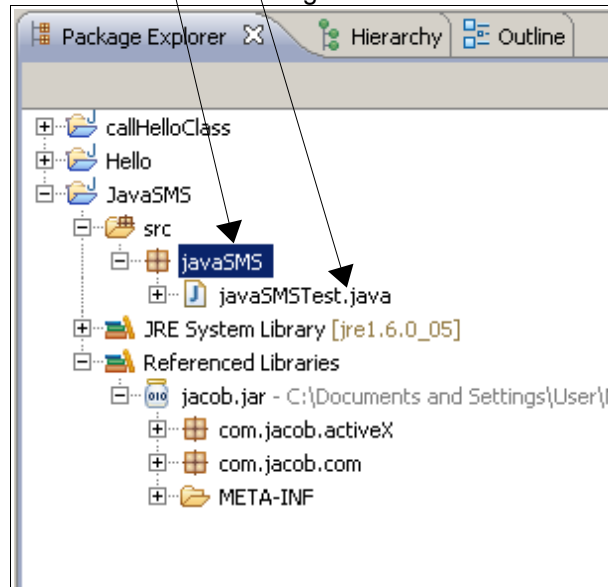
5. Go to "Libraries" tab
6. Click on "Add External JARs" button
7. Select the file "jacob.jar" (that was unzip from "jacob-1.14.3.zip")
8. The final outcome should be similar to figure right
9. Click on "Finish" button



# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

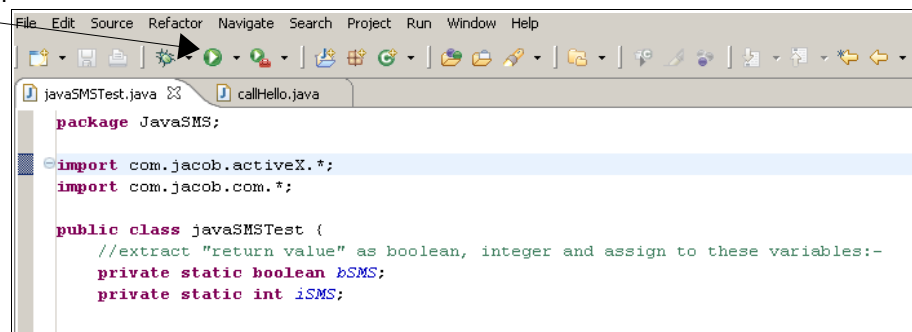
10. Add a new package, named it "javaSMS"
11. Add a new class, named it "javaSMSTest.java"
12. And your final outcome should be similar to figure below:



13. Copy the Java sample code (refer to page 10; also available in CD) and paste to "javaSMSTest.java"
14. In the sample code, change the COM port from "2" to the respective COM port:  

```
Variant SMSAPIReturnValue = SMSAPIJava.invoke("ModemInit", 2);
```
15. In the Java code, change the "0163311600" to the intended recipient's mobile number:  

```
SMSAPIJava.setProperty("ToNumber", "0163311600");
```
16. Click "Run" button

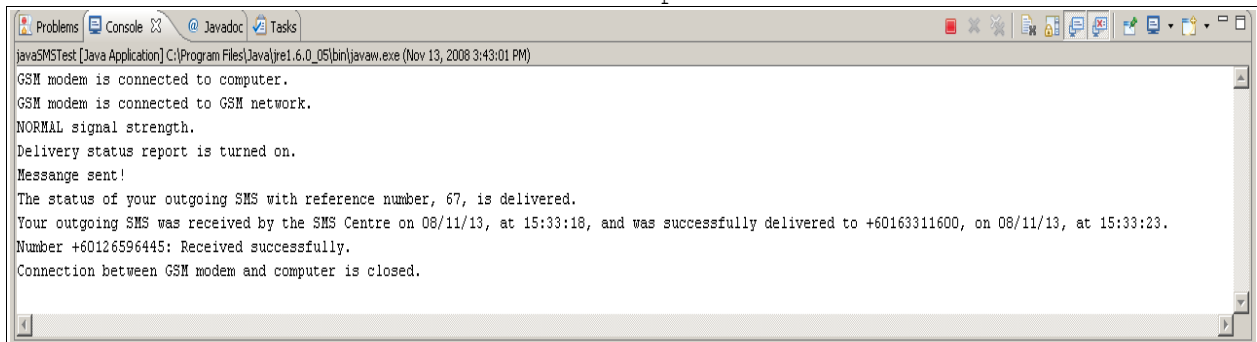


# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### 17. And you will see the result in "Console" tab:

```
GSM modem is connected to computer.  
GSM modem is connected to GSM network.  
NORMAL signal strength.  
Delivery status report is turned on.  
Messange sent!  
The status of your outgoing SMS with reference number, 67, is delivered.  
Your outgoing SMS was received by the SMS Centre on 08/11/13, at 15:33:18,  
and was successfully delivered to +6016xxxxxxx, on 08/11/13, at 15:33:23.  
Number +6012xxxxxxx: Received successfully.  
Connection between GSM modem and computer is closed.
```



### 18. You have successfully used Java to:

- a) connect GSM modem with PC/server;
- b) check signal strength;
- c) send out SMS;
- d) get delivery status report;
- e) read SMS; and
- f) disconnect GSM modem with PC/server



# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### JAVA QUICK GUIDE

#### SENDING SMS

```
//----- send SMS -----
//set value of property; "ToNumber" is the recipient's number;
// "ToMessage" is the SMS to be send to the recipient
SMSAPIJava.setProperty("ToNumber", "0163311600");
SMSAPIJava.setProperty("ToMessage", "Hello from JAVA. Test DSR 6");
//call API to send out SMS, the return value of the API call is
assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("SendSMS");
//----- end: send SMS -----
```

#### READING SMS

```
//+++++++ read SMS ++++++
//call API to read incoming SMS, the return value of the API call is
assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("ReadSMS");
//since the return value of "ReadSMS" is a boolean, so extract the
return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();
//if there is new SMS, then get property
if (bSMS)
{
    //property "MN" is the sender's number
    String InNumber = SMSAPIJava.getPropertyAsString("MN");
    //property "MSG" is the sender's message
    String InMessage = SMSAPIJava.getPropertyAsString("MSG");
    System.out.println("Number " + InNumber + ": " + InMessage );
}
else
{
    System.out.println("No incoming SMS!");
}
//+++++++ end: read SMS ++++++
```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

### JAVA SAMPLE CODE

```
package javaSMS;

import com.jacob.activeX.*;

public class javaSMSTest {
    //extract "return value" as boolean, integer and assign to these
    variables:-
    private static boolean bSMS;
    private static int iSMS;

    /**
     * @param args
     */
    public static void main(String[] args) {

        //construct an object -- "SMSAPIJava", a Java wrapper for SMS class
        in "MobitekSMSAPI5.dll"
        ActiveXComponent SMSAPIJava = new ActiveXComponent
        ("MobitekSMSAPI5.SMS");

        //connect GSM modem to computer
        //call API to connect GSM modem to computer at COM port no. 2
        //the return value of the API call is assigned to
        "SMSAPIReturnValue"
        Variant SMSAPIReturnValue = SMSAPIJava.invoke("ModemInit", 2);
        //since the return value of function "ModemInit" is an integer, so
        extract the return value as an integer and assign to "iSMS"
        iSMS = SMSAPIReturnValue.getInt();
        switch (iSMS)
        {
            case 0 : System.out.println("GSM modem is NOT connected to
            computer!"); return;
            case 1 : System.out.println("GSM modem is connected to
            computer."); break;
            case 2 : System.out.println("GSM modem is NOT connected to
            computer because a PIN is required!"); return;
            case 3 : System.out.println("GSM modem is NOT connected to
            computer because wrong PIN is entered!"); return;
            case 4 : System.out.println("GSM modem is NOT connected to
            computer because SIM card is blocked by network operator!"); return;
            case 5 : System.out.println("GSM modem is NOT connected to
            computer because SIM card has problem!"); return;
        }
        //end: connect GSM modem to computer

        //check connection between GSM modem and GSM
        network
        //call API to check connection with GSM network, the return value of
        the API call is assigned to "SMSAPIReturnValue"
        SMSAPIReturnValue = SMSAPIJava.invoke("ConnectToGSM");
    }
}
```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

```
//since the return value of "ConnectToGSM" is a boolean, so extract
the return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();
if (bSMS)
{
    System.out.println("GSM modem is connected to GSM network.");
}
else
{
    System.out.println("GSM modem is NOT connected to GSM
network!");
}
//!!!!!!!!!!!! end: check connection between GSM modem and GSM
network !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

//~~~~~ check signal strength of GSM network ~~~~~
//call API to check signal strength of GSM network, the return value
of the API call is assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("GetSignalStrength");
//since the return value of "GetSignalStrength" is an integer, so
extract the return value as an integer and assign to "iSMS"
iSMS = SMSAPIReturnValue.getInt();
switch (iSMS)
{
    case -1 : System.out.println("Unable to get signal
strength!"); break;
    case 0 : System.out.println("NO signal strength!"); break;
    case 1 : System.out.println("WEAK signal strength!"); break;
    case 2 : System.out.println("NORMAL signal strength!"); break;
    case 3 : System.out.println("STRONG signal strength!"); break;
}
//~~~~~ end: check signal strength of GSM networki
~~~~~

//##### turn delivery status report on #####
//call API to turn delivery status report on
//the return value of the API call is assigned to
"SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("DeliveryReportOn");
//since the return value of "DeliveryReportOn" is a boolean, so
extract the return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();
if (bSMS)
{
    System.out.println("Delivery status report is turned on.");
}
else
{
    System.out.println("Delivery status report is NOT turned
on!");
}
//##### end: turn delivery status report on #####
```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

```
//----- send SMS -----
//set value of property; "ToNumber" is the recipient's number;
"ToMessage" is the SMS to be send to the recipient
SMSAPIJava.setProperty("ToNumber", "0163311600");
SMSAPIJava.setProperty("ToMessage", "Hello from JAVA. Test DSR 6");
//call API to send out SMS, the return value of the API call is
assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("SendSMS");
//since the return value of "SendSMS" is a boolean, so extract the
return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();
if (bSMS)
{
    System.out.println("Messange sent!");

    //try 3 times
    for (int i=1; i<=3; i++)
    {
        //call API to get delivery status report, the return
value of the API call is assigned to "SMSAPIReturnValue"
        SMSAPIReturnValue =
SMSAPIJava.invoke("GetDeliveryReport");
        //since the return value of "GetDeliveryStatusReport" is
a boolean, so extract the return value as a boolean and assign to "bSMS"
        bSMS = SMSAPIReturnValue.getBoolean();

        if (bSMS)
        {
            //when "GetDeluveryStatusReport = True", then get
value of properties
            int DRStatus =
SMSAPIJava.getPropertyAsInt("DRStatus");

            String DRMNRecipient =
SMSAPIJava.getPropertyAsString("DRMNRecipient");
            String DRMsgRef =
SMSAPIJava.getPropertyAsString("DRMsgRef");
            String DRFDate =
SMSAPIJava.getPropertyAsString("DRFDate");
            String DRFTime =
SMSAPIJava.getPropertyAsString("DRFTime");
            String DRRDate =
SMSAPIJava.getPropertyAsString("DRRDate");
            String DRRTIME =
SMSAPIJava.getPropertyAsString("DRRTIME");

            if (DRStatus == 1)
            {
                //System.out.println("The status of your
outgoing SMS with reference number, " + DRMsgRef + ", is " + DRStatus + ".");
                System.out.println("The status of your
outgoing SMS with reference number, " + DRMsgRef + ", is delivered.");
                System.out.println("Your outgoing SMS was
```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

```

received by the SMS Centre on " + DRRDate + ", at " + DRRTIME + ", and was
successfully delivered to " + DRMNRecipient + ", on " + DRFDate + ", at " +
DRFTime + "."");
    }
    else
        if (DRStatus == 0)
        {
            //System.out.println("The status of
your outgoing SMS with reference number, " + DRMsgRef + ", is " + DRStatus +
"."");
            System.out.println("The status of your
outgoing SMS with reference number, " + DRMsgRef + ", is not delivered.");
            System.out.println("Your outgoing SMS
was received by the SMS Centre on " + DRRDate + ", at " + DRRTIME + ", and was
NOT successfully delivered to " + DRMNRecipient + "."");
        }
        else
            if (DRStatus == 2)
            {
                //System.out.println("The status of
your outgoing SMS with reference number, " + DRMsgRef + ", is " + DRStatus +
"."");
                System.out.println("The status of your
outgoing SMS with reference number, " + DRMsgRef + ", is unknown.");
                System.out.println("Your outgoing SMS
was received by the SMS Centre on " + DRRDate + ", at " + DRRTIME + ", and NO
status is available.");
            }

        break;
    }
    else
    {
        System.out.println("No delivery status report
available!");
    }
}
else
{
    System.out.println("Messange NOT sent!");
}
//----- end: send SMS -----

//+++++++ read SMS ++++++
//call API to read incoming SMS, the return value of the API call is
assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("ReadSMS");
//since the return value of "ReadSMS" is a boolean, so extract the
return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();

```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

## GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

```
//if there is new SMS, then get property
if (bSMS)
{
    //property "MN" is the sender's number
    String InNumber = SMSAPIJava.getPropertyAsString("MN");
    //property "MSG" is the sender's message
    String InMessage = SMSAPIJava.getPropertyAsString("MSG");
    System.out.println("Number " + InNumber + ": " + InMessage );
}
else
{
    System.out.println("No incoming SMS!");
}

//+++++ end: read SMS +++++

//----- close connection between GSM modem and computer -----
//call API to close connection between GSM modem and computer, the
return value of the API call is assigned to "SMSAPIReturnValue"
SMSAPIReturnValue = SMSAPIJava.invoke("ModemClose");
//since the return value of "ModemClose" is a boolean, so extract
the return value as a boolean and assign to "bSMS"
bSMS = SMSAPIReturnValue.getBoolean();
if (bSMS) System.out.println("Connection between GSM modem and
computer is closed." );
else System.out.println("Connection between GSM modem and computer
CANNOT be closed!" );
}

//----- end: close connection between GSM modem and computer -----

}
```

# DEVELOPING A SMS GATEWAY BY USING JAVA AND COM

GUIDE FOR SYSTEM INTEGRATOR & SOFTWARE DEVELOPER

## TERMS AND CONDITIONS ON THE USE OF THE JAVA SAMPLE CODE

1. This sample code is to be used with **SMS API**.
2. You are allowed to use, modify and distribute this code for free.
3. This sample code is provided on "as-is" basis. **No warranty nor support** is provided on the use of the code.
4. By using this code, you assume all the risk and responsibility.
5. MOBITEK System Sdn. Bhd. expressly disclaim any liability and warranty.

## TERMS AND CONDITIONS ON SUPPORT

1. **No support is provided** that includes but not limited to:-
  - a) No support nor training on how to write a Java application;
  - b) No support on how to use JACOB; and
  - c) No support on how to.