

## How\_to\_add\_contacts\_using\_JSR\_75

Following code snippets show how to add contacts using JSR 75.

```
/**
 * PIMWrite Class
 * This sample demo generates the random names and numbers and it will add to the phonebooks.
 * This code will be useful, if we want to add more contacts to our phonebooks for testing.
 * if we need to see the performance, of PhoneBooks contacts writing and reading speed
 * This code will be useful
 * @author Senthil
 *
 */
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.util.*;

public class PIMWrite extends MIDlet implements CommandListener
{
    private Command PIMWriteCmd,exitCommand,generateName;
    PIMImpl pimImpl;
    private static Random random = new Random();
    static String[] names;

    public static Display display;
    public static Form result;
    private String sp,si,s1;
    private String[] strJSR = new String[12];
    private long l;
    int noOfContacts;

    TextField textfield = new TextField("Enter No of Contacts: ", "", 3, TextField.NUMERIC);

    /*Constructs an Object
    */
    public PIMWrite() {

        display = display.getDisplay(this);

        PIMWriteCmd = new Command("PIMWrite", Command.SCREEN, 0);

        exitCommand = new Command("Exit", Command.SCREEN, 0);

        new Form("Write Contacts To PIM");
        result.append(textfield);
        System.out.println(" noOfContacts: "+noOfContacts);

        addCommand(PIMWriteCmd);
        addCommand(exitCommand);
        setCommandListener(this);
    }

    public void startApp() throws MIDletStateChangeException {
        display.setCurrent(result);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {

```

## How\_to\_add\_contacts\_using\_JSR\_75

```
}

public void commandAction(Command c, Displayable s)
{
    if (c == exitCommand)
    {
        destroyApp(false);
        notifyDestroyed();
    }

else if(c == PIMWriteCmd)
{
    = Integer.parseInt(textfield.getString());
    createConatactName
    System.out.println(" noOfContacts: "+noOfContacts);
    //Start the PIM thread and write contacts .
    = new PIMImpl();pimImpl
    start();        pimImpl.

    }
}

public void createConatactName()
{
    = new String[noOfContacts];
for (int i = 0; i < noOfContacts; i++){
    [i] = PIMWrite.generateName()+" "+PIMWrite.generateName();
System.out.println(" Names :"+ names[i]+ " i "+i);
}
}

    public static String generateName()
{
return generateName(4, 7, true);
}

/** This method generates the random names
 *
 * @param minLength Minimum length of name
 * @param maxLength Max length of name
 * @param capitalize Captitalize
 * @return name
 */
    public static String generateName(int minLength, int maxLength, boolean capitalize)
{

int limit = (random.nextInt(maxLength - minLength + 1)) + minLength;
StringBuffer s = new StringBuffer(limit);
for (int i = 0; i < limit; i++)
{
char c = (char)((random.nextInt(26)) + 'a');
append(c);        s.
}
if (capitalize && (s.length() > 0)) s.setCharAt(0, Character.toUpperCase(s.charAt(0)));
return s.toString();
}
}
```

## PIM Implementation class

```

/**
 * PIM Implementation class
 */
import javax.microedition.pim.*;
import java.util.*;
import javax.microedition.lcdui.*;
import javax.microedition.lcdui.Command;

public class PIMImpl extends Thread
{
    static Alert check;

    StringItem si;
    //-----Phone Book-----

    int noOfRecords;
    public static boolean pimTemp = false;
    Random randPh = new Random();

    public PIMImpl()
    {

    }

    public void run()
    {
        ();    writeData
    }

    public synchronized void writeData()
    {

    String ph[] = new String[PIMWrite.names.length];
        for (int i=0;i<PIMWrite.names.length ;i++ )
        {

    String r = Long.toString(randPh.nextLong());
    if(r.startsWith("-"))
    {
        = r.replace('-', '9');    r
    }
    [i] =(r.length()>10)?(phsubstring(0,10)): (r);

    //    System.out.println(" PH :"+ph[i] +"i:"+i);
    }

    int tc = 0;
        ;PIM pim
        try{

            ContactList contacts;
            ContactList num;
            = PIM.getInstance();

```

## How\_to\_add\_contacts\_using\_JSR\_75

```
try {
    pim.openPIMList (PIM.CONTACT_LIST, PIM.READ_WRITE);
} catch (PIMException e) {
    // An error occurred
    return;
}
for(int i=0;i<PIMWrite.names.length;i++)
{
    Contact contact = contacts.createContact();
    String[] name = new String[contacts.stringArraySize(Contact.NAME)];

    // System.out.println("i = "+i);
    if (contacts.isSupportedField(Contact.TEL))
    {
        addStringOfContact(Contact.TEL, Contact.ATTR_MOBILE, ph[i]);
    }

    if (contacts.isSupportedArrayElement(Contact.NAME, Contact.NAME_GIVEN))
        [Contact.NAME_GIVEN] = PIMWrite.names[i];
        addStringArray(Contact.NAME, PIMItem.ATTR_NONE, name);

    try {
        commit();
    } catch (PIMException e) {
        // An error occurred
    }

    result.addElement();
    String msg=" Contacts Adding :"+i;
    = new StringItem("",msg);
    result.append(i);
}

    check new Alert("Added","Added ",null,AlertType.CONFIRMATION);
    setTimeout(Alert.FOREVER);
    displayPIMCurrent(check);

    try {
        close();
    } catch (PIMException e) {
    }

} catch (Exception e) {
    String er = e+" "; tc="+tc;
    = new Alert("Response",er,null,AlertType.CONFIRMATION );
    setTimeout(Alert.FOREVER);

    PIMDisplay.setCurrent(check);
}
// return true;
}
}
```