How to store Data in RMS

When we execute certain applications on our mobile devices, often, it is necessary to store data for possible reuse. Java ME this allows to make this through the API RMS (Record Management System).

RMS is the manager of the database, and largely consists of a set of record stores. Record store, in turn, is nothing more than a set of records composed of two fields. An identifier (number) and a data field (an array of bytes). The record stores are linked directly to the MIDlet suite and not with the MIDlets, so if you want each one of the MIDlet has his owns record store, it is necessary to use different names, since these are case sensitive. When the MIDlet suite is removed, the same applies to the record stores.

The simple example following shows how to use the RMS API:

```
import javax.microedition.rms.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.io.*;
public class ReadWriteRMS
        extends MIDlet implements CommandListener {
    private Display display;
    private Alert alert;
    private Form form;
    private Command exit;
    private Command start;
    private Command delete;


    private RecordStore recordstore= null;
    RecordEnumeration recEnum;

    public ReadWriteRMS() {
        display = Display.getDisplay(this);
        exit = new Command("Exit", Command.EXIT, 1);
        start = new Command("Start", Command.SCREEN, 1);
        delete = new Command("Delete",Command.SCREEN,2);
        form = new Form("Mixed Record");
        form.addCommand(exit);
        form.addCommand(start);
        form.addCommand(delete);

        form.setCommandListener(this);
    }
    public void startApp() {
        display.setCurrent(form);
    }
    public void pauseApp() {
    }
    public void destroyApp( boolean unconditional ) {
    }
    public void commandAction(Command command, Displayable displayable) {
        if (command == exit) {
            destroyApp(true);
            notifyDestroyed();
        } else if (command == start) {
            try {
```


(C) Copyright Nokia 2014. All rights reserved.

```
                recordstore = RecordStore.openRecordStore("myRecordStore", true );



            } catch (Exception error) {
                error.printStackTrace();
                alert = new Alert("Error Creating", error.toString(), null,
AlertType.WARNING);
                alert.setTimeout(Alert.FOREVER);
                display.setCurrent(alert);
            }
            try {
                byte[] outputRecord;


                String outputString[] =
{"why_you","Changed","my_names","andAdded","your","name"};
                int outputInteger[] = {25,21,20,23,34,34};
                boolean outputBoolean[] = {true,false,true,true,false,true};

                ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
                DataOutputStream outputDataStream = new DataOutputStream(outputStream);

                for(int i=0; i<outputString.length;i++) {
                    outputDataStream.writeUTF(outputString[i]);
                    outputDataStream.writeBoolean(outputBoolean[i]);
                    outputDataStream.writeInt(outputInteger[i]);
                    outputDataStream.flush();
                    outputRecord = outputStream.toByteArray();
                    recordstore.addRecord(outputRecord, 0, outputRecord.length);
                }

                outputStream.reset();
                outputStream.close();
                outputDataStream.close();
            } catch ( Exception error) {
                alert = new Alert("Error Writing",
                        error.toString(), null, AlertType.WARNING);
                alert.setTimeout(Alert.FOREVER);
                display.setCurrent(alert);
            }

// Read records from RMS.........
            try {
/* String inputString = null;
int inputInteger = 0;
boolean inputBoolean = false; */

                StringBuffer buffer = new StringBuffer();

                byte[] byteInputData = new byte[100];
                ByteArrayInputStream inputStream = new
ByteArrayInputStream(byteInputData);
                DataInputStream inputDataStream = new DataInputStream(inputStream);
                int numrec = recordstore.getNumRecords();
                System.out.println(" Num rec "+numrec);
                for(int i=1;i<=numrec;i++) {
```


(C) Copyright Nokia 2014. All rights reserved.

```
// recordstore.getRecord(i);
                    recordstore.getRecord(i,byteInputData,0);
                    buffer.append(inputDataStream.readUTF());
                    buffer.append("\n");
                    buffer.append(inputDataStream.readBoolean());
                    buffer.append("\n");
                    buffer.append(inputDataStream.readInt());
                    buffer.append("\n");
                    alert = new Alert("Reading", buffer.toString(),
                            null, AlertType.WARNING);
                    alert.setTimeout(Alert.FOREVER);
                    display.setCurrent(alert);
                }
/* recEnum = recordstore.enumerateRecords(null,null,false);
while (recEnum.hasNextElement())
{
recordstore.getRecord(recEnum.nextRecordId(),byteInputData,0);
buffer.append(inputDataStream.readUTF());
buffer.append("\n");
buffer.append(inputDataStream.readBoolean());
buffer.append("\n");
buffer.append(inputDataStream.readInt());
buffer.append("\n");
alert = new Alert("Reading", buffer.toString(),
null, AlertType.WARNING);
alert.setTimeout(Alert.FOREVER);
display.setCurrent(alert);
}
 */
                inputStream.close();
                inputDataStream.close();
            } catch (Exception error) {
                alert = new Alert("Error Reading",
                        error.toString(), null, AlertType.WARNING);
                alert.setTimeout(Alert.FOREVER);
                display.setCurrent(alert);
            }
            try {
                recordstore.closeRecordStore();
            } catch (Exception error) {
                alert = new Alert("Error Closing",
                        error.toString(), null, AlertType.WARNING);
                alert.setTimeout(Alert.FOREVER);
                display.setCurrent(alert);
            }
        }

        else if( command == delete) {

            if (RecordStore.listRecordStores() != null) {
                try {
                    RecordStore.deleteRecordStore("myRecordStore");
                } catch (Exception error) {
                    alert = new Alert("Error Removing",
                            error.toString(), null, AlertType.WARNING);
                    alert.setTimeout(Alert.FOREVER);
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    display.setCurrent(alert);
                }
            }
        }
    }
}
```


(C) Copyright Nokia 2014. All rights reserved.