How to store and fetch images from RMS

In this article is presented a code example for loading and storing images using the Record Management System.

Overview

Record Management System (RMS) can be used to any kind of information if the storage limits of the phone are respected. The data is always stored as a byte array and a proper byte stream need to be provided. In this code example is showed how you can store and load images using RMS.

Storing an image

The image byte array can be created using an input stream, a byte array output stream and a byte vector. It is necessary to feed the input stream by an image. At the end, `imageBytes` will contains the image as a byte array, suitable for RMS. See the code snippet below:

```
int ch;
byte[] imageBytes;
InputStream inputStream = null;
inputStream = getClass().getResourceAsStream("/lumia800.jpg");
ByteArrayOutputStream byteOutStream = new ByteArrayOutputStream();
while ((ch = inputStream.read()) != -1) {
  byteOutStream.write(ch);
}
imageBytes = byteOutStream.toByteArray();
record.addRecord(imageBytes, 0, imageBytes.length);
```

Loading an image

A byte array input stream is required in the loading operation to transform the byte array in an image, like below:

```
InputStream is = new ByteArrayInputStream(record.getRecord(IMG_RMS_ID));
Image im = Image.createImage(is);
```

Application code

A complete reference application is below. After starting, it is possible to load an image from resources (lumia800.jpg) using the "Load" menu option. This image will be showed in the main screen and can be saved again via "Store" menu option. The image record ID inside RMS is fixed and defined by the constant IMG_RMS_ID. More images can be stored if you use different record IDs.

```
/**
 * This MIDlet stores the images into the RMS, and fetches the images from RMS and
displays them.
 */
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
import javax.microedition.rms.*;
import java.io.*;

public class StoreImage extends MIDlet implements CommandListener {

    private Image image = null;
    private Display display = null;
```


(C) Copyright Nokia 2014. All rights reserved.

```
    private Command storeImage = null;
    private Command loadImage = null;
    private Command exit = null;
    private Alert alert = null;
    private Form form = null;
    private RecordStore record = null;
    // imagem position at data store
    private final int IMG_RMS_ID = 1;

    public StoreImage() {
        form = new Form("RMS Image");
        storeImage = new Command("Store", Command.SCREEN, 1);
        loadImage = new Command("Load", Command.SCREEN, 1);
        alert = new Alert("Image Stored", "Image Stored",null,AlertType.INFO);
        exit = new Command("Exit", Command.EXIT, 1);
        form.addCommand(storeImage);
        form.addCommand(loadImage);
        form.addCommand(exit);
        form.setCommandListener(this);
        display = Display.getDisplay(this);
    }

    /*
     * This method opens the Record store
     */
    public RecordStore openRecordStore() {
        try {
            record = RecordStore.openRecordStore("StoreImage", true);
        } catch (Exception e) {
            log("Err in open RS");
        }
        return record;
    }

    /* This method is used for storing the images into the RMS
     * lumia800.jpg" file should be in the resources dir
     */
    private void storeImage() {
        InputStream inputStream = null;
        try {
            inputStream = getClass().getResourceAsStream("/lumia800.jpg");
            ByteArrayOutputStream byteOutStream = new ByteArrayOutputStream();
            int ch;
            byte[] imageBytes;
            while ((ch = inputStream.read()) != -1) {
                byteOutStream.write(ch);
            }
            imageBytes = byteOutStream.toByteArray();
            openRecordStore();
            // Try to record reusing position IMG_RMS_ID.
            // If not available, create a new record. It is assumed that
            // the store is empty so the first item ID to be created
            // will be 1. A better handler for IDs may be required here.
            try {
                record.setRecord(IMG_RMS_ID, imageBytes, 0, imageBytes.length);
            }
```


(C) Copyright Nokia 2014. All rights reserved.

```
            catch (Exception e2){
                record.addRecord(imageBytes, 0, imageBytes.length);
            }
            display.setCurrent(alert);
            log("Image stored to RMS");
            inputStream.close();
        } catch (Exception e) {
            log("Err in Add Image to RMS" + e);
        }
    }

    /*
     * Retrieves the Image from the RMS and displays.
     */
    public void loadImage() {
        try {
            openRecordStore();
            InputStream is = new ByteArrayInputStream(record.getRecord(IMG_RMS_ID));
            Image im = Image.createImage(is);
            form.deleteAll();
            form.append(im);
        } catch (Exception e) {
        }
    }

    public void startApp() {
        display.setCurrent(form);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean b) {
    }

    public void commandAction(Command c, Displayable d) {
        if (c == exit) {
            destroyApp(true);
            notifyDestroyed();
        } else if (c == storeImage) {
            storeImage();
        } else if (c == loadImage) {
            loadImage();
        }
    }

    public void log(String msg) {
        System.out.println("Msg: " + msg);
    }
}
```

Source Code

The source code is available here. It was tested using Netbeans 7.0.1 and SDK 1.1 for Series 40 but it should work with any Java ME enabled phone with JSR 118 (Mobile Information Device Profile 2.0).

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.