

Use_sms_as_data_bearer

Moving data over GPRS or other data connections is typically the least cost method. But sometimes, especially in emerging markets, users may not have access to a data service.

Here is the full source code for a Midlet that supports user-generated localization for an application to connect people to community radio stations.

Video demo and discussion about the full app here.

When a typical data bearer is not present, the Midlet falls back to SMS as the bearer. This code snippet shows how.

```
/*
 *
 * Written by Gergely Herenyi e-mail: gergely.herenyi at nokia.com
 */

package com.nokia.radiostation.connectivity;

import com.nokia.radiostation.languages.Language;
import java.io.ByteArrayOutputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import javax.microedition.io.Connector;
import javax.wireless.messaging.BinaryMessage;
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.MessageListener;
import javax.wireless.messaging.TextMessage;

/**
 * SMS handling class
 */
public class SMSBearer implements MessageListener {

    /** Message connection for sending/receiving SMSs */
    MessageConnection messconn;
    /** Reference to the parent connection object*/
    Connection parent;
    /** The maximum data an SMS can transfer in bytes */
    public static final int SMS_SIZE = 133;

    /**
     * Constructor for SMSBearer
     *
     * @param parent the parent connection object
     */
    public SMSBearer(Connection parent) {
        this.parent = parent;
        register();
    }

    /**
     * Registers for incoming messages on port 1976
     */
    public void register() {
        try {
            messconn = (MessageConnection) Connector.open("sms://:1976");
            messconn.setMessageListener(this);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Use_sms_as_data_bearer

```
}

/**
 * Handles incoming messages
 *
 * @param conn MessageConnection where the message arrives
 */
public void notifyIncomingMessage(MessageConnection conn) {
    if (conn == messconn) {
        try {
            TextMessage msg = (TextMessage) messconn.receive();
            String address = msg.getAddress();
            if (address.startsWith("sms://")) {
                address = address.substring(6);
            }
            if (address.lastIndexOf(':') != -1) {
                address = address.substring(0, address.lastIndexOf(':'));
            }
            String request = msg.getPayloadText();
            processRequest(request, address);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

/**
 * Processes and replies to a request
 *
 * @param request the request, which is received
 * @param number the number where the reply should be sent to
 */
public void processRequest(String request, String number) {
    if (request.startsWith("RADIO")) {
        sendSMS(number, parent.parent.stations[parent.parent.selectedStation].toByte());
    } else if (request.startsWith("GET_LANGUAGES")) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        DataOutputStream os = new DataOutputStream(baos);
        try {
            os.writeUTF(Language.getLanguages()[Language.getSelectedLanguage()]);
        } catch (IOException e) {
            e.printStackTrace();
        }
        sendSMS(number, baos.toByteArray());
    } else if (request.startsWith("GET_LANGUAGE_")) {
        sendSMS(number, Language.binaryLanguage);
    }
}

/**
 * Sends an SMS
 *
 * @param number where the SMS is sent to
 * @param data the message to be sent
 */
public void sendSMS(String number, byte[] data) {
    String port = "1976";
    int parts = data.length / (SMS_SIZE - 2) + 1;
    int offset = 0;
    for (int i = 0; i < parts; i++) {
        int length = Math.min(data.length - offset, SMS_SIZE - 2);
        byte[] dataToSend = new byte[2 + length];
    }
}
```

Use_sms_as_data_bearer

```
dataToSend[0] = (byte) i;
dataToSend[1] = (byte) parts;

System.arraycopy(data, offset, dataToSend, 2, length);
offset += length;
try {
    MessageConnection msgConn = (MessageConnection) Connector.open("sms://" + number);
    BinaryMessage msg = (BinaryMessage) msgConn.newMessage(MessageConnection.BINARY_M
    msg.setPayloadData(dataToSend);
    msgConn.send(msg);
    msgConn.close();
} catch (IOException e) {
    e.printStackTrace();
} catch (SecurityException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
}
}
}
```