Sending a multipart MMS using Java ME

Overview

This code snippet demonstrates how to send a multipart MMS message. In the snippet, the user first enters a destination address to a text field. He or she may also fetch a phone number from the address book. After that, the user enters a subject for the message. Then, by selecting `Send`, an image is attached to the message and the message is sent to the specified address.

In the code, the following message parameters are set:

- Destination address (specified by the user)
- Subject of the message (specified by the user)
- Priority (set to normal)
- Message content (a JPEG image loaded from a resource).

This is a complete example MIDlet, but the most interesting methods (methods regarding multipart message handling) are `handleSendCommand()`, `prepareMessage()`, `createMsgPart()`, and `sendMMS()`.

Source file

```
import java.io.IOException;
import java.io.InputStream;
import java.io.InterruptedIOException;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Form;
import javax.microedition.lcdui.TextField;
import javax.microedition.midlet.MIDlet;
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.MessagePart;
import javax.wireless.messaging.MultipartMessage;
import javax.wireless.messaging.SizeExceededException;
```

```
public class MMSMIDlet extends MIDlet implements CommandListener {
    private Command sendCommand;
    private Command exitCommand;
    private Form mainForm;
    private TextField mmsAddress;
    private TextField mmsSubject;
    private MessageConnection connection;

    /**
     * Constructor. Constructs the object and initializes displayables.
     */
    public MMSMIDlet() {
        mainForm = new Form("MMS Example");

        mmsAddress = new TextField("Address", null, 40,
            TextField.PHONENUMBER);
        mainForm.append(mmsAddress);

        mmsSubject = new TextField("Subject", null, 255,
```


(C) Copyright Nokia 2014. All rights reserved.

```
            TextField.PLAIN);
        mainForm.append(mmsSubject);

        sendCommand = new Command("Send", Command.ITEM, 0);
        mainForm.addCommand(sendCommand);

        exitCommand = new Command("Exit", Command.EXIT, 0);
        mainForm.addCommand(exitCommand);

        mainForm.setCommandListener(this);
    }

    /**
     * From MIDlet.
     * Called when the MIDlet is started.
     */
    public void startApp() {
        // The initial display is the main form
        Display.getDisplay(this).setCurrent(mainForm);
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to enter the Paused state.
     */
    public void pauseApp() {
        // No implementation required
    }

    /**
     * From MIDlet.
     * Called to signal the MIDlet to terminate.
     * @param unconditional whether the MIDlet has to be unconditionally
     * terminated
     */
    public void destroyApp(boolean unconditional) {
        if (connection != null) {
            try {
                // Close the message connection
                connection.close();
            } catch (IOException ex) {
                // TODO: Exception handling
            }
        }
    }

    /**
     * From CommandListener.
     * Called by the system to indicate that a command has been invoked on a
     * particular displayable.
     * @param command the command that was invoked
     * @param displayable the displayable where the command was invoked
     */
    public void commandAction(Command command, Displayable displayable) {
        if (command == exitCommand) {
            // Exit the MIDlet
            destroyApp(true);
```


(C) Copyright Nokia 2014. All rights reserved.

```
            notifyDestroyed();
        } else if (command == sendCommand) {
            handleSendCommand();
        }
    }

    private void handleSendCommand() {
        try {
            // Open the connection
            connection = (MessageConnection)Connector.open("mms://:MyApp");
        } catch (IOException ex) {
            // TODO: Exception handling
        }

        MultipartMessage message = prepareMessage();
        sendMMS(message);
    }

    /**
     * Sets the parameters for the multipart message.
     */
    private MultipartMessage prepareMessage() {
        // Prepare the multipart message
        MultipartMessage message = (MultipartMessage)connection.newMessage(
            MessageConnection.MULTIPART_MESSAGE);

        // Set the destination address
        String address = "mms://" + mmsAddress.getString();
        message.setAddress(address);

        // Set the subject
        String subject = mmsSubject.getString();
        message.setSubject(subject);

        // Set the priority
        String priority = "normal";  // "high", "normal" or "low"
        message.setHeader("X-Mms-Priority", priority);

        // Set the message part
        try {
            MessagePart messagePart = createMsgPart();
            message.addMessagePart(messagePart);
        } catch (SizeExceededException ex) {
            // TODO: Exception handling
        } catch (IOException ex) {
            // TODO: Exception handling
        }

        return message;
    }

    /**
     * Constructs a MessagePart which can be added to a MultipartMessage.
     * @return the constructed MessagePart
     * @throws javax.wireless.messaging.SizeExceededException if the contents
     *   is larger than the available memory or supported size for the message
```


(C) Copyright Nokia 2014. All rights reserved.

```
     *   part
     * @throws java.io.IOException if the resource cannot be read
     */
    private MessagePart createMsgPart() throws SizeExceededException,
            IOException {
        String imageContentID = "image01";
        String imageContentLocation = "old_nokia_logo.jpg";
        String jpgMIME = "image/jpeg";
        InputStream imageContent = getClass().getResourceAsStream(
            imageContentLocation);
        MessagePart messagePart = new MessagePart(imageContent, jpgMIME,
                imageContentID, imageContentLocation, null);
        return messagePart;
    }

    /**
     * Sends a multipart message.
     */
    private void sendMMS(final MultipartMessage message) {
        // Send the message on its own thread of execution
        Thread messageThread = new Thread() {
            public void run() {
                try {
                    connection.send(message);
                    mainForm.append("Message sent.");
                } catch (InterruptedIOException ex) {
                    // TODO: Exception (e.g. timeout) handling
                } catch (IOException ex) {
                    // TODO: Exception (e.g. network failure) handling
                } catch (IllegalArgumentException ex) {
                    // TODO: Exception (e.g. too big or otherwise invalid
                    // message) handling
                } catch (SecurityException ex) {
                    // TODO: Exception (e.g. insufficient permissions) handling
                }
            }
        };
        messageThread.start();
    }
}
```

Postconditions

The MIDlet sends an MMS message to the address specified by the user. The message consists of the subject text (specified by the user) and an image (loaded from a resource).

See also

- Listening asynchronously for incoming SMS messages in Java ME
- How to send a message to a given port as SMS with Java ME

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.