



Introduction of MMS in J2ME

August 28, 2006

TECHNICAL ARTICLE

Introduction of MMS in J2ME

By
MOTODEV Staff

MMS (Multimedia Message Service) enables users of multimedia messaging-enabled handsets able to bundle text, music, pictures and video in one package and send it to the other handsets or email addresses. MMS is one of the most popular value-added services for communication operators. JSR205 WMA (Wireless Messaging API) 2.0 provides support for MMS.

This article focuses on sending and receiving MMS with J2ME APIs in JSR205. It introduces the structure of multimedia messages, describes the basic steps to sending and receiving multimedia messages and provides an example to illustrate the sending and receiving of a simple MMS message.

Structure of Multimedia Messages

Multimedia messages contain a message header and multipart message body. The MIME type [RFC 2045-7] is used to encode and package the multipart in multimedia message and is thus compatible with email.

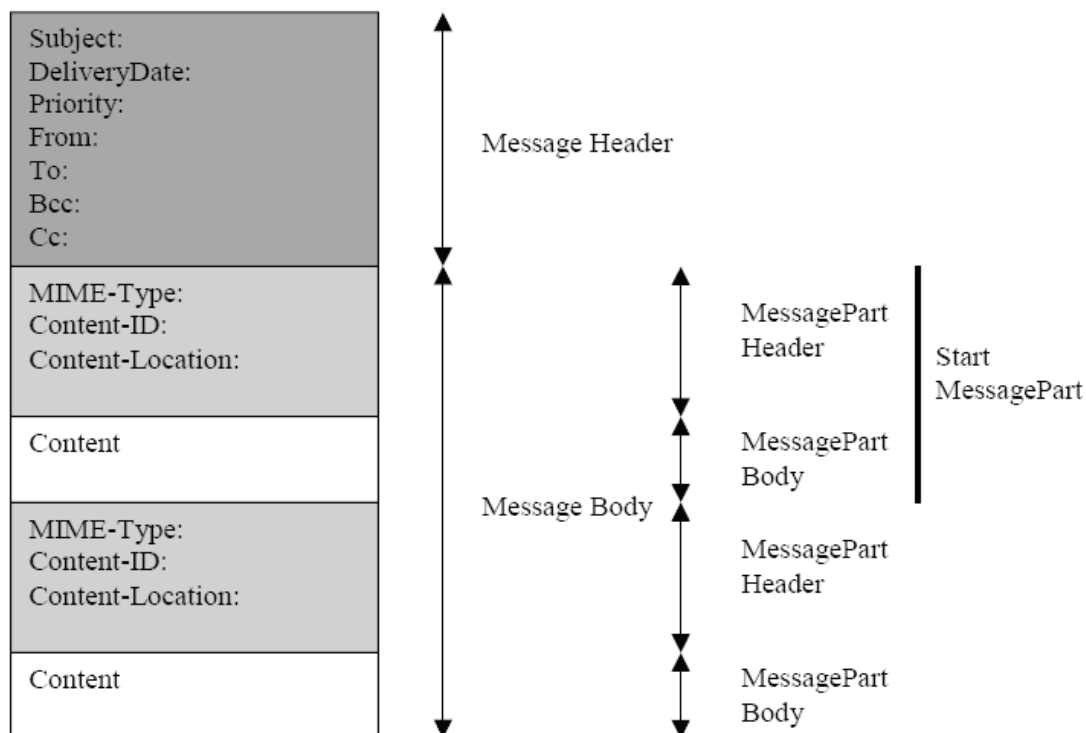


Figure 1: Multimedia Message Structure

The message header includes the basic properties of the message:

- A MIME-Type indicating the content in the MessagePart, (e.g., image/jpeg)
- A possible Content-Location that can be used as a Synchronized Multimedia Integration Language (SMIL) reference
- A Content-Identifier (Content-ID) identifies the content (used as a start parameter and as a SMIL reference)

SMIL is a XML based presentation W3C standard which defines the layout and playing time for MMS Multimedia Objects. A subset of SMIL is used as the presentation language for MMS. The SMIL file is also packaged as a MessagePart.

Key steps to sending and receiving multimedia messages

MMS connection setup is based on the Generic Connection Framework. Usually handsets supporting JSR205 have a native MMS application, which is the default application to edit, send and receive multimedia messages. To ensure success in sending and receiving multimedia messages between J2ME applications, the following additional parameters are added to the Content-Type header field as additional Content-Type parameters: *Application-ID*; *Reply-To-Application-ID*.

To send a Multimedia Message:

1. Set up an MMS connection:

```
String mmsUrl;
mmsUrl = "mms://+13507102543"; //send to default mms application
mmsUrl = "mms://+13507102543:com.mot.MMSEExample"; //send to J2me MMS application, and
com.mot.MMSEExample is the application ID.
mmsUrl = "mms://:com.mot.MMSEExample"; // server-side connection to send or receive messages
mmsURL = mms://test@mot.com; // send to email address
MessageConnection connection =(MessageConnection)Connector.open(mmsUrl);
```

2. Build the multipart message

```
//Create message and set header properties
MultipartMessage message = (MultipartMessage) connection.newMessage(
    MessageConnection.MULTIPART_MESSAGE);
message.setAddress(cardMessage.getAddress());
message.setSubject(cardMessage.getSubject());
//Create message partes
byte[] imageBytes = cardMessage.getImageBytes();
MessagePart cardPart= new MessagePart(imageBytes,0,imageBytes.length,"image/png",
    "card","card image",null);
byte[] textMsgBytes = cardMessage.getTextMessage().getBytes();
MessagePart textPart= new MessagePart(textMsgBytes,0,textMsgBytes.length,"text/plain",
    "message","message text","UTF-8");
//Add parts to the message body
message.addMessagePart(cardPart);
message.addMessagePart(textPart);
```

3. Send the MMS message in an independent thread

```
connection.send(message);
```

To receive MMS message, open a server connection first, then listen for the incoming message incident.

1. Open a server connection

```
String mmsUrl = "mms://:com.mot.MMSExample";
try {
    connection =(MessageConnection)Connector.open(mmsUrl);
}
catch(Exception e){
    ...
}
```

2. Set message listener, receive and process the message when it comes

```
connection.setMessageListener(this);

public void notifyIncomingMessage(MessageConnection connection){
    cdMessage = mgManager.receiveMessage();
    ...
}
```

3. Receive and get message parts

```
MultipartMessage message = (MultipartMessage)connection.receive();
MessagePart[] parts = message.getMessageParts();
cardMessage = new CardMessage();
cardMessage.setSubject(message.getSubject());
for(int i=0;i<parts.length;i++){
    MessagePart part = parts[i];
    if(part.getMIMEType().equals("image/png")){
        cardMessage.setImage(part.getContent());
    }
    else if(part.getMIMEType().equals("text/plain")){
        cardMessage.setTextMessage(new String(part.getContent()));
    }
}
```

4. Set permissions for open connection

```
javax.microedition.io.Connector.mms
```

5. Set permissions for send and receive operations

```
javax.wireless.messaging.mms.send
javax.wireless.messaging.mms.receive
```

6. Push Registry

```
MIDlet-Push-1: mms://: com.mot.MMSExample, com.mot.MMSExample,MMSReceiveMidlet
```

Example

The [sample included with this article](#) is easy to use: launch the MMSReceive Midlet on one handset; then start the MMSSendMidlet on another handset, input the destination address and subject, and send the message to the receiver; the receiver will take the message and display it.

Conclusion

With the JSR205 APIs, we are able to send/receive/parse multimedia messages with a few lines of code. It's a powerful API, but making it cooperate with native MMS applications and operator MMS services can still pose problems.

References

SMIL <http://www.w3c.org/AudioVideo>

WMA2.0 <http://www.jcp.org/en/jsr/detail?id=205>

Sample Message <http://www.waynet.cn/conch/list.asp?unid=1044>

Appendix

Sample Multimedia Message

MIME Object	
Date: Fri, 14 Dec 2001 15:13:21 -0500 (EST) X-Mms-Expiry: Thu, 20 Dec 2001 21:52:25 -0500 (EST) X-Mms-Delivery-Time: Wed, 19 Dec 2001 21:52:25 -0800 (PST) X-Mms-Priority: Normal X-Mms-Sender-visibility: Show X-Mms-Message-Class: Personal X-Mms-Delivery-report: yes X-Mms-3GPP-MM3-Version: 4.3.0 From: mms-email-sample@sample-email.ericsson.com To: +12345@mmc.sample-mms.ericsson.com Subject: Sample SMIL Mime-Version: 1.0 Content-Type: multipart/related; boundary="---- = MIME_Boundary_MMS_Reference_Sample_001"	MIME Headers
"----= MIME_Boundary_MMS_Reference_Sample_001"	MIME Start boundary
Content-ID: <index.smil> Content-Type: application/smil; charset="US-ASCII"; name=index.smil <smil> <head> <layout><root-layout/> <region id="region1_2" top="0" left="0" height="100%" width="100%"/> <region id="region1_1" top="0" left="0" height="100%" width="100%"/> </layout> </head> <body> <par dur="2000ms"> <text src="Cid:image" region="region1_2"/> </par> <par dur="2000ms"> </par> </body> </smil>	SMIL document
"----= MIME_Boundary_MMS_Reference_Sample_001"	MIME boundary
Content-Type: text/plain; name=mms.txt	Text Document

Content-Transfer-Encoding: 7bit Content-ID: <txt> Content-Disposition: attachment; filename=mms.txt This is a sample MMS message, for use as a reference.	
"----- MIME Boundary MMS Reference Sample_001"	MIME boundary
Content-Type: image/gif; name=sample.gif Content-Transfer-Encoding: base64 Content-ID: <image> Content-Disposition: attachment; filename=sample.gif R0IGODlhZQBQAPcAAOYeACIUpyAtb/XNjlo8RqoUFHSa2ZCKjHiCWrf/2iQ1J m66Yeq4cXm/6ys <entire encoded image not shown for the sake of brevity> kBiJkjiJFiJlniJmJiJmriJnNiJnviJoBiKogh5AQEAOW==	Image
"----- MIME Boundary MMS Reference Sample_001"	MIME boundary