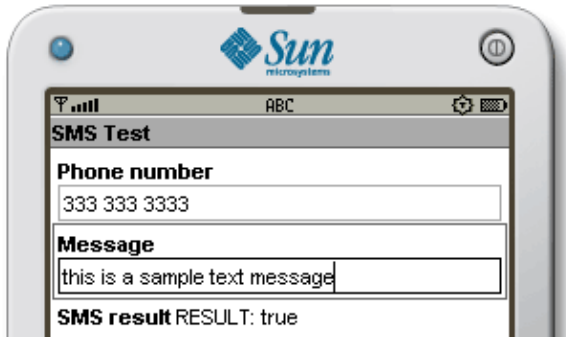




WMAPI allows Java ME applications to access messaging functionalities, as sending and receiving SMS and MMS messages. This article explains how to use it to send a simple text message.



Contents

- [1 Source code](#)
 - ◆ [1.1 Download](#)
 - ◆ [1.2 Notes](#)
- [2 Related resources](#)

Source code

The following Java ME tip explains a method of sending text messages . In the program TextMessage interface represents a text message. The setPayloadText() method sets the characters in the message.

```
public boolean sendSms(String number, String message){
    boolean result = true;
    try {
        //sets address to send message
        String addr = "sms://" + number;
        // opens connection
        MessageConnection conn = (MessageConnection) Connector.open(addr);
        // prepares text message
        TextMessage msg =
        (TextMessage) conn.newMessage(MessageConnection.TEXT_MESSAGE);
        //set text
        msg.setPayloadText(message);
        // send message
        conn.send(msg);
        conn.close();
    } catch (SecurityException se) {
        // probably the user has not allowed to send sms
        // you may want to handle this differently
        result = false;
    } catch (Exception e) {
        result = false;
    }
}
```

How_to_Send_Text_SMS_in_Java_ME

```
    return result;  
}
```

Download

You can download a sample MIDlet showing the code presented in this article here:

[Media:HowToSendTextSMSMIDlet.zip](#)

Notes

- Remember that a single message has a maximum size (140 bytes, which translates to 160 7-bit characters). If your message is too long to fit in one message it can be splitted to a maximum of three SMS, this is handled automatic by the WMA implementation.
- **Potentially-blocking operations**, like sending a text message, **should always be performed in separate threads**, to avoid blocking the main MIDlet thread.

Related resources

- [WMAPI JavaDocs](#)
- [Wireless messaging section on Java ME Developer's Library](#)