Converting a resource into a byte array in Java ME

Version Hint

Windows Phone: [[Category:Windows Phone]]
[[Category:Windows Phone 7.5]]
[[Category:Windows Phone 8]]

Nokia Asha: [[Category:Nokia Asha]]
[[Category:Nokia Asha Platform 1.0]]

Series 40: [[Category:Series 40]]
[[Category:Series 40 1st Edition]] [[Category:Series 40 2nd Edition]]
[[Category:Series 40 3rd Edition (initial release)]] [[Category:Series 40 3rd Edition FP1]] [[Category:Series 40 3rd Edition FP2]]
[[Category:Series 40 5th Edition (initial release)]] [[Category:Series 40 5th Edition FP1]]
[[Category:Series 40 6th Edition (initial release)]] [[Category:Series 40 6th Edition FP1]] [[Category:Series 40 Developer Platform 1.0]] [[Category:Series 40 Developer Platform 1.1]] [[Category:Series 40 Developer Platform 2.0]]

Symbian: [[Category:Symbian]]
[[Category:S60 1st Edition]] [[Category:S60 2nd Edition (initial release)]] [[Category:S60 2nd Edition FP1]] [[Category:S60 2nd Edition FP2]] [[Category:S60 2nd Edition FP3]]
[[Category:S60 3rd Edition (initial release)]] [[Category:S60 3rd Edition FP1]] [[Category:S60 3rd Edition FP2]]
[[Category:S60 5th Edition]]
[[Category:Symbian^3]] [[Category:Symbian Anna]] [[Category:Nokia Belle]]

Overview

This code snippet demonstrates how to convert a resource into a byte array. The snippet assumes that the resource is included in the JAR file, which is used to distribute the application.

Example: To include a file to the JAR package in NetBeans 6.1, do as follows:

1. In the **Projects** window (select **Window** > **Projects** if it's not visible), open the project node for the project to which you want to add the file.
2. Right-click on the **Resources** node under the project and choose **Add Folder**. The Add Folder dialog box opens.
3. Select the folder where the resource is (for example, `dist/images`).
4. Click **Open**. The folder is added to the project.
5. Clean and build the project (right-click on the project node and select **Clean & Build**).
6. The resource can be obtained in the code, for example, like this:

```
InputStream inputStream = getClass().getResourceAsStream("image.jpg");
// ...
```

Source

```
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
```

```
/**
 * Constructs a byte array and fills it with data that is read from the
 * specified resource.
 * @param filename the path to the resource
 * @return the specified resource as a byte array
 * @throws java.io.IOException if the resource cannot be read, or the
 *   bytes cannot be written, or the streams cannot be closed
```


(C) Copyright Nokia 2014. All rights reserved.

```
 */
private byte[] obtainByteData(String filename) throws IOException {
    InputStream inputStream = getClass().getResourceAsStream(filename);
    ByteArrayOutputStream outputStream = new ByteArrayOutputStream();

    // Read bytes from the input stream in bytes.length-sized chunks and write
    // them into the output stream
    for (int readBytes = inputStream.read(); readBytes >= 0; readBytes =
inputStream.read())
        outputStream.write(readBytes);

    // Convert the contents of the output stream into a byte array
    byte[] byteData = outputStream.toByteArray();

    // Close the streams
    inputStream.close();
    outputStream.close();

    return byteData;
}
```

The method above can be used, for example, with multipart messages. In the following, a `MessagePart` object is constructed from a byte array which represents raw data from a JPEG image:

```
import javax.wireless.messaging.MessagePart;
import javax.wireless.messaging.SizeExceededException;

// ...

/**
 * 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
 *   part
 * @throws java.io.IOException if the byte data cannot be obtained
 */
private MessagePart createMsgPart() throws SizeExceededException,
        IOException {
    String imageContentID = "image01";
    String imageContentLocation = "image.jpg";
    String jpgMIME = "image/jpeg";
    // Convert the image into a byte array and construct a message part using
    // it
    byte[] imageContent = obtainByteData(imageContentLocation);
    MessagePart messagePart = new MessagePart(imageContent, jpgMIME,
            imageContentID, imageContentLocation, null);
    return messagePart;
}
```

Postconditions

A specified resource is converted into a byte array.

See also

(C) Copyright Nokia 2014. All rights reserved.

- Sending a multipart MMS using Java ME

(C) Copyright Nokia 2014. All rights reserved.