

Known Issues In The Nokia 6600 MIDP 2.0 Implementation

Version 1.7; February 3, 2005

Java™

NOKIA

Copyright © 2005 Nokia Corporation. All rights reserved.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

The information in this document is provided "as is," with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

Change History

December 16, 2003	Version 1.0	Initial document release
March 9, 2004	Version 1.1	Section 2.12 changed. Sections 2.14 and 2.15 added.
April 7, 2004	Version 1.2	Chapter 1 and Sections 2.1, 2.4, 2.8, 2.9, 2.10, 2.13, 2.14, and 2.17 updated. Sections 2.11, 2.12, 2.18, 2.19, 2.20, 2.21, 2.22, and 2.23 added.
May 11, 2004	Version 1.3	Sections 2.24 and 2.25 added.
November 10, 2004	Version 1.4	Sections 2.26, 2.27, and 2.28 added.
December 7, 2004	Version 1.5	Chapter 1 updated. Section 2.29 and Chapter 4 added.
December 22, 2004	Version 1.6	Section 2.30 added.
February 3, 2005	Version 1.7	Section 2.31 added.

Contents

1	Introduction	6
2	Known Issues	7
2.1	Garbage Collecting	7
2.2	Reading Resources from a JAR File	7
2.3	Thread Usage	8
2.4	Display.setCurrent(Alert alert, Displayable nextDisplayable) Method	9
2.5	Default Encoding Is ISO-8859-1 Instead of UTF-8	9
2.6	Getting the Current Time Takes the TimeZone into Account.....	9
2.7	UTF-8 Characters in MIDlet-Name Attribute.....	10
2.8	Command Label Is Always “Exit”, if Command.EXIT Type Is Used	10
2.9	Using Backlight and Vibration.....	10
2.10	Changing the FullScreenMode of Canvas to True.....	10
2.11	Commands in Full Screen.....	11
2.12	Menu Is Displayed (Open) and Displayable Is Changed	11
2.13	Using MIDlet.platformRequest(String URL).....	11
2.14	Using Ticker in a Form	11
2.15	Using Nokia UI API.....	11
2.16	MIDI Sounds in MIDlets Running in Background.....	12
2.17	Switching Displayables when the Running MIDlet Is Obscured by a System Screen	12
2.18	Adding a MIDlet Icon	12
2.19	Size of the MIDlet Icon	13
2.20	Adding Commands to Alert Screens.....	13
2.21	VideoControl.setDisplayLocation(x,y) Does Not Work Correctly.....	13
2.22	Installing Signed MIDlets	13
2.23	Using flashBacklight() Method	13
2.24	Series 60 Profile Settings and Playing Sounds in MIDlets	14
2.25	Video Stream Resize Fails with USE_GUI_PRIMITIVE	14
2.26	MIDlet Installation Problem	14
2.27	Mobile Media API-Captured Image	15
2.28	SSL Handshake Is Done for Every HTTPS Connection that Is Made	15
2.29	Display.setCurrentItem() Does Not Work.....	15
2.30	Using Bluetooth Serial Port in MIDlets	15
2.31	The Root Certificate for Java Verified Is Missing.....	16
3	Supported Interfaces/Packages in the Nokia 6600 MMAPi Implementation	17
3.1	Player Content Types.....	17
3.2	Capture Locators.....	17

3.3	Video Snapshot Image Types	18
3.4	Supported Controls	18
3.5	Excluded Features	18
3.6	VideoControl Notes	19
3.7	Time Bases.....	19
3.8	RTP Streaming.....	19
4	Evaluate This Document	20

1 Introduction

This document presents known issues related to the MIDP 2.0 implementation in the first commercial software versions of the Nokia 6600 device. The software version numbers are 3.42.1 and 4.09.1. The software version can be checked by typing `*#0000#`. Note that new software versions will not be released for the Nokia 6600 device.

This document is intended to give guidance for developers who want to make sure their MIDP 1.0 MIDlets run in the Nokia 6600. There are some basic differences in the behavior of the Nokia 6600 (MIDP 2.0) compared to Nokia MIDP 1.0 devices.

2 Known Issues

This chapter lists known issues related to the MIDP 2.0 implementation of the Nokia 6600 device. The issues are not presented in any particular order.

2.1 Garbage Collecting

Description:

Calling the `System.gc()` method results in extreme slowness and jamming. In Monty 1.0 VM, garbage collection is different and every time `System.gc` is called, the entire memory is really cleared. This is an extremely slow process!

Solution:

Do not call the `System.gc` method at all, or call the `System.gc()` garbage collecting method only in non-time-critical situations, such as screen transitions, state transitions, pause states, etc. If the `System.gc()` method is used, it is recommended to add a short delay (~20-50 ms) after the method call to ensure the sufficient time for the garbage collection, as in the following example:

```
System.gc ();

Thread.sleep(delay); //delay = 20-50 ms
```

2.2 Reading Resources from a JAR File

Description:

The MIDP 2.0/Monty implementation for the `InputStream.read(byte[])` method in the Nokia 6600 device takes a literal reading of the definitions of `java.io.InputStream.read(bytes[] b)` and `java.io.InputStream.read(byte[] b, int off, int len)`, which indicate that these functions might NOT read the number of bytes equal to the size of the array argument (`b.length`) before returning, and will indicate the number of bytes that were successfully read as a return value. This is basically because it is a more optimized approach for the implementation.

NOTE: Because all other `InputStream` classes (for example, `DataInputStream` and `ByteArrayInputStream`) are subclasses of `InputStream`, this issue is the same for ALL `InputStream` operations.

Solution:

The Java™ programmer (in this case for MIDlets on the Nokia 6600 device, but this is generally the best practice) should always loop until reading is complete and not assume a resource will be read in one go; see the following examples of wrong and right code:

WRONG

```
InputStream in = getClass().getResourceAsStream(resource);
if (in == NULL) return;
DataInputStream din = new DataInputStream(in);
int bytes = din.read(data);
din.close();
Image I = Image.createImage(data, 0, offset);
```

RIGHT

```
InputStream in = getClass().getResourceAsStream(resource);
```

```

if (in == null) return;
DataInputStream din = new DataInputStream(in);
int bytes = 0;
int offset = 0;
while (true) {
    bytes = din.read(data, offset, data.length - offset);
    offset += bytes;
    if (bytes == -1 || offset >= data.length) {
        // may need to handle error condition here!
        break;
    }
}
din.close();
// now it's safe to assume data is already fully read (or error
// occurred) for example, if it was image data..
Image I = Image.createImage(data, 0, offset);

```

2.3 Thread Usage

Description:

If a MIDlet is running any threads when exiting, the running threads may keep on hanging, thus reserving memory even after the MIDlet has exited.

Solution:

Before exiting an application with the Exit command or in some other graceful exit situation, all of the threads have to be killed. This can be done either in the `destroyApp()` method or in some other user-specified method that is called before exiting the MIDlet, for example, something like the following:

```

public void commandAction(Command c, Displayable d) {
    if (c == exitCommand) {
        // exit the MIDlet
        iRunning = false;
    }
}

public void run()
{
    while( iRunning )
    {
        repaint();
        serviceRepaints();
        try{ Thread.sleep(100); }catch( InterruptedException e ){}
    }
    MidletMain.quitApp();
}

```

2.4 Display.setCurrent(Alert alert, Displayable nextDisplayable) Method

Description:

The method public void setCurrent(Alert alert, Displayable nextDisplayable) requests an Alert to be shown; after the Alert is dismissed, nextDisplayable is shown. Currently, this works incorrectly. The title of nextDisplayable is shown on the screen, but otherwise the display contents are not updated. After this, no display changes are possible.

Solution:

First, set the Displayable A, then set the Alert B (by using the setCurrent() method). Then add the CommandListener to the Alert. When the Alert is dismissed, you get the event. You can then change the Displayable in the Listener. Another option is to use thread and its Sleep() method and the Displayable.setCurrent() method to change the displayable, when the Alert is dismissed.

This issue has been corrected in software version 4.09.1.

2.5 Default Encoding Is ISO-8859-1 Instead of UTF-8

Description:

Default encoding is ISO-8859-1 instead of UTF-8.

Solution:

When using data, which should be UTF-8, do not assume the default charset is UTF-8 but define it explicitly. For example:

```
String s = new String(byteArray, "utf-8");
```

2.6 Getting the Current Time Takes the TimeZone into Account

Description:

CLDC 1.0 specification requires the method System.currentTimeMillis() to return the time in UTC (GMT).

Because some earlier phone models supported only time zone UTC, MIDlet programmers did not always take the difference between UTC and local time into account. MIDlets did work correctly assuming that the returned value was in local time, without even checking time zone settings.

In the Nokia 6600 device such existing MIDlets might behave differently, using UTC instead of local time as expected in many use cases.

Solution:

Use a Calendar object to get the local time.

```
Calendar calendar = Calendar.getInstance() ; // uses default TimeZone
Calendar.setTime( new Date( System.currentTimeMillis() ) ;
```

Note: The phone user can set the time zone as follows:

1. By turning the automatic time update on. This is a network service and is not supported in all networks.
2. By setting the time zone manually in the Clock application or under Settings / Date & Time.

MIDlet programmers should be aware that Java applications are just about the only application in the phone that uses the time zone. Thus, users might have incorrect time zone settings in their phone without being aware of it.

2.7 UTF-8 Characters in MIDlet-Name Attribute

Description:

If UTF-8 “special” characters (for example, ä, ö, å, or Chinese characters) are used in the MIDlet-Name attribute, the MIDlet will not start.

Solution:

Don't use non-ASCII characters in the MIDlet-Name attribute.

2.8 Command Label Is Always “Exit”, if Command.EXIT Type Is Used

Description:

If the Command.EXIT type is used, the Command label is always “Exit”, regardless of the label set in the Command constructor.

Solution:

Command type Command.CANCEL is mapped to the right soft key, like Command.EXIT type. It can be used instead.

This issue has been corrected in software version 4.09.1.

2.9 Using Backlight and Vibration

Description:

There are methods for using backlight and vibration both in the Nokia UI API (DeviceControl.flashLights(long duration) and DeviceControl.startVibra (int freq, long duration)) and in MIDP 2.0 (flashBacklight(int duration) and vibrate(int duration)). It is recommended not to use the Nokia UI API methods.

Solution:

The MIDP 2.0 methods should be used. In software versions 3.42.1 and 4.09.1, only backlight is supported.

2.10 Changing the FullScreenMode of Canvas to True

Description:

If the FullScreenMode of Canvas is changed to true, the Commands in the Options menu don't have any effect. If any soft key is pressed, the Options menu is shown, but choosing a Command does nothing.

Solution:

No solutions exist for software version 3.42.1 or 4.09.1.

2.11 Commands in Full Screen

Description:

Commands can be added to full screen but they do not work.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software version 4.09.1.

2.12 Menu Is Displayed (Open) and Displayable Is Changed

Description:

If the menu is displayed (open) and the displayable is changed, the old commands are shown but if you try to execute the commands, the new commands (from the new displayable) are executed.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software version 4.09.1.

2.13 Using MIDlet.platformRequest(String URL)

Description:

This method requests the device to handle (for example, display or install) the indicated URL. For example, if the URL is of the form <http://www.nokia.com>, the WAP browser should be launched. For a URL containing a telephone number, a voice call should be initiated. However, this does not work in software version 3.42.1.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software version 4.09.1.

2.14 Using Ticker in a Form

Description:

When using Ticker in a Form and changing a new Form into display, the Ticker stays visible.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software version 4.09.1.

2.15 Using Nokia UI API

Description:

Some methods of Nokia UI API don't work. Also, all the audio and video formats that are supported in MIDP 2.0 and Mobile Media API (MMAPI) are not supported in Nokia UI API.

Solution:

It is not recommended to use the Nokia UI API anymore in Nokia 6600. It is included in order to maintain backward compatibility, but basically the same features and more are implemented in MIDP 2.0 and MMAPI classes. For more information, please read Chapter 3.

2.16 MIDI Sounds in MIDlets Running in Background**Description:**

The MIDI sound playback is not stopped if the MIDlet is moved to the background. If the Application key is pressed while the MIDlet is running, the MIDlet is left in the background. If a MIDI sound has been playing, it continues playing even if the MIDlet is not visible. There seems to be no easy way to track if any of the Displayables are visible or not.

Solution:

This problem exists in Screens such as Form and List, but not in Canvas. In Canvas it is easy to track whether the Canvas is visible or not by using the `hideNotify()` and `showNotify()` methods. Such methods don't exist for Screens.

2.17 Switching Displayables when the Running MIDlet Is Obscured by a System Screen**Description:**

If the `setCurrent()` method has been called in the `Canvas.hideNotify()` method and the red phone button is pressed, the phone desktop is quickly shown and the application is brought to the foreground showing the screen that was chosen in the `setCurrent()` method instead of showing the phone desktop.

Setting the current Displayable in the `hideNotify()` method will make the application come to the foreground. Thus, if the user presses the Applications key to access the Applications menu, the menu is shown only quickly and the application is brought to the foreground. The user must press the Applications key again to see the menu.

Solution:

It is not necessary to set the current Displayable in the `hideNotify()` method. The `showNotify()` method can be used instead.

2.18 Adding a MIDlet Icon**Description:**

The JAD and manifest file attribute for MIDlet suite ('MIDlet-Icon') does not have an effect. Only the default icon is shown in the menu.

Solution:

The MIDlet icon is shown in the device's menu only if the icon for MIDlet ('MIDlet-1') has been defined. Thus, if the JAD and manifest files contain the following row, the icon is shown in the menu.

```
MIDlet-1: TestMIDlet, /icon.png, TestMIDlet
```

2.19 Size of the MIDlet Icon

Description:

If the MIDlet icon size (width and height) is too large, the icon is shown very small in the menu.

Solution:

The maximum size for the MIDlet icon is 29x29. If the icon is larger, it is shown small in the device's menu.

2.20 Adding Commands to Alert Screens

Description:

MIDP 2.0 enables adding of Commands to Alert Screens. It is possible to add Commands to Alert Screens in Nokia 6600, but the Commands are not shown correctly. Only the default command 'OK' is shown with the Alert components.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software version 4.09.1.

2.21 VideoControl.setDisplayLocation(x,y) Does Not Work Correctly

Description:

VideoControl.setDisplayLocation(x,y) does not work in the Nokia 6600 software version 3.42.1. The method should play the video file in a Canvas at location (x, y), but the video is played at the default location (0, 0) instead.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software release 4.09.1.

2.22 Installing Signed MIDlets

Description:

Signed MIDlets are not supported in the Nokia 6600 software version 3.42.1. The only supported domain is untrusted domain. Thus only MIDP 1.0 MIDlets and untrusted MIDP 2.0 MIDlets can be installed.

Solution:

No solutions exist for software version 3.42.1, but this issue has been corrected in software release 4.09.1, where 3rd party domain will be supported.

2.23 Using flashBacklight() Method

Description:

The method flashBacklight() does not work correctly. After the flashing effect the screen stays dimmed.

Solution:

No solutions exist. The normal backlight can be adjusted from the screen brightness in the phone settings.

2.24 Series 60 Profile Settings and Playing Sounds in MIDlets**Description**

Sound playback is based only on the “Warning tones” setting. MIDlet sounds are always played back when the “Warning tones” setting has been set to “ON.”

If the Silent profile has been selected, normally no MIDlet sounds are played. However, if the “Warning tones” setting has been set to “ON” in the Silent profile, the sounds are played.

Solution

No solutions exist.

2.25 Video Stream Resize Fails with USE_GUI_PRIMITIVE**Description**

MMAPI's `VideoControl.setDisplaySize()` method is not working correctly when the `USE_GUI_PRIMITIVE` mode is used.

According to the MMAPI Javadocs for `VideoControl.setDisplaySize()`: "If the video mode is set to `USE_GUI_PRIMITIVE`, scaling the video will also scale the GUI object." In the Nokia 6600, the video is resized properly but the GUI object that contains the video stream is not resized against the video.

Solution

No solutions exist.

2.26 MIDlet Installation Problem**Description**

In very rare cases, installing a MIDlet to the Nokia 6600 smartphone via OTA or Web Push may for some reason fail in the middle of the installation process.

If installing a MIDlet to the Nokia 6600 device via OTA or Web Push for some reason fails in the middle of the installation process, the half-installed MIDlet cannot be removed and the same MIDlet cannot be installed to the device. However, this happens very rarely.

This concerns software version 4.09.1.

Solution

No solution exists for Nokia 6600. Fixed in the Series 60 Platform 2nd Edition, Feature Pack 1 (that is, the Nokia 6620 and Nokia 7610 devices).

2.27 Mobile Media API-Captured Image

Description

A Mobile Media API-captured image has poor quality when compared to an image captured with the native camera application.

The Mobile Media API 1.0 has been implemented in the Nokia 6600 device. The maximum resolution of the captured image is 176 x 132.

This concerns software version 4.09.1.

Solution

In Nokia 6600 device, the best quality snapshot can be taken when the image size is smaller than 176 x 132 and viewfinder is used. In the Nokia 6620 and Nokia 7610 devices the default image size is the biggest possible resolution.

2.28 SSL Handshake Is Done for Every HTTPS Connection that Is Made

Description

When using HTTPS connections, the Nokia 6600 device seems to perform a new Secure Sockets Layer (SSL) handshake for every connection that is made to the same server. This slows down significantly the performance of the HTTPS connection.

This concerns software versions 3.42.1 and 4.09.1.

Solution

Have the connection open all the time or, alternatively, decrease the amount of created connections.

2.29 Display.setCurrentItem() Does Not Work

Description

The `Display.setCurrentItem()` method does not work in software version 4.09.1. Calling this method does not change the focus and make the selected Item visible.

Solution

No solution exists.

2.30 Using Bluetooth Serial Port in MIDlets

Description

Reading incoming stream does not work properly. When the Bluetooth Serial Port Profile (RFCOMM protocol) is used, reading incoming stream is sometimes interrupted with `IOException`. The `read()` method of the `InputStream` class may throw an `IOException` if the other end has closed the stream while the `InputStream` is still reading it.

Another issue is that the `InputStream` cannot use a bigger array than 512 bytes when reading stream. A bigger byte array may crash the MIDlet.

Run, for example, the `BtSppEcho` example application to reproduce the issue.

Solution

Solution to the `IOException` issue: Ensure that the `InputStream.read()` has finished before closing the output stream in the other end.

Solution to the stream size issue: Do not use a bigger byte array than 512 bytes. Read the input stream in a loop or keep the whole length of the stream under 512 bytes.

Generic workaround to both issues: Use L2CAP connection instead of SPP (RFCOMM), if appropriate. This is feasible when you have access to the L2CAP protocol in both ends of your application (for example, connections between two devices).

2.31 The Root Certificate for Java Verified Is Missing

Description

The Java Verified testing and certification program uses a UTI root certificate (also known as "GeoTrust CA for UTI"), which is missing from the Nokia 6600 device. When this root certificate is missing, the Java Verified tested applications cannot be installed. The shipment of the Nokia 6600 started before the certificate was available.

Solution

Java Verified applications can be installed if the following attributes are removed from the JAD file: `MIDlet-Certificate-1-1` and `MIDlet-Jar-RSA-SHA1`. Without these lines the MIDlet is treated as untrusted and it thereby gives a warning message during the installation. This makes the application appear as if it were unsigned, but it is possible to install the application similarly as any unsigned application. When the `MIDlet-Certificate-1-1` and `MIDlet-Jar-RSA-SHA1` are removed, the `MIDlet-Permissions` attribute becomes redundant and it can be removed as well.

3 Supported Interfaces/Packages in the Nokia 6600 MMAPI Implementation

3.1 Player Content Types

Mobile Media API implementation supports the following content types:

- **audio/x-tone-seq** = tone sequence
- **audio/wav** = wav audio format, but player cannot be created from InputStream with this content type
- **audio/x-wav** = wav audio format
- **audio/au** = au audio format
- **audio/basic** = raw audio format
- **audio/x-au** = au audio format, but same problem as with audio/wav
- **audio/amr** = amr audio format
- **audio/amr-wb** = amr wb audio format
- **audio/sp-midi** = sp-midi audio format (extended midi format)
- **audio/midi** = midi audio format
- **video/mp4** = mpeg4 video format
- **video/mpeg4** = mpeg4 video format, same problem as with audio/wav
- **video/3gpp** = 3gpp video format
- **application/vnd.rn-realmedia** = real media video format

Additional note: Due to native software limitations, some of the “duplicate” content types cannot be used, for example, you can use wav with type audio/x-wav, but not with audio/wav. Certain duplicated MIME types cannot be used when creating a Player from an InputStream. Those MIME types work when a Player is created from a URI locator. In this case you don't need to specify the MIME type, but the MIME type is specified at the server side, for instance, `Player player = Manager.createPlayer(String url)`.

3.2 Capture Locators

Mobile Media API implementation supports recording of audio and capturing of video snapshots (photos) from the camera. The following forms of capture locators are supported:

- `capture://audio` for recording of audio
- `capture://video` for capturing of snapshots from the camera

The video parameters width, height and encoding are not supported because video cannot be recorded. The video parameter fps is supported and can take values between 1 and 30.

For audio recording, you should use: `"capture://audio?encoding=pcm&signed=unsigned"`. Other workarounds: 8khz & 16bit `"capture://audio?rate=8000&bits=16"`. You do not have “the mickey mouse” -effect with 8 or 16 kHz. Or simply: `"capture://audio?rate=8000"` (bits are not necessary).

3.3 Video Snapshot Image Types

If a video snapshot is taken (by VideoControl's method 'getSnapshot'), the supported image types are "encoding=png", "encoding=bmp", and "encoding=jpeg". The image type may optionally include width and height parameters (which default to 160x120):

- videoControl.getSnapshot(null); // default PNG, 160x120
- videoControl.getSnapshot("encoding=bmp"); // BMP, 160x120
- videoControl.getSnapshot("width=80&height=60"); // default PNG, 80x60
- videoControl.getSnapshot("encoding=bmp&width=80&height=60"); // BMP, 80x60

If values are specified for width and height, both must be specified, not just width or height. In this case, the image is scaled to the requested width and height. If the aspect ratio requested does not match that of the default 160x120 (that is, 4:3), the resulting image will be distorted. The maximum size that can be captured depends on the free heap memory available.

3.4 Supported Controls

Mobile Media API implementation supports the following controls for each content type:

MIME type	Controls
Audio/x-tone-seq	VolumeControl, StopTimeControl, ToneControl
Audio/wav, Audio/au, Audio/amr	VolumeControl, StopTimeControl, RecordControl
Audio/x-wav, Audio/basic, Audio/x-au, Audio/amr-wb, Audio/sp-midi, Audio/midi	VolumeControl, StopTimeControl
Video/mp4, Video/mpeg4, Video/3gpp, Application/vnd.rn-realmedia	VolumeControl, StopTimeControl, VideoControl
Video capture	VideoControl

3.5 Excluded Features

The following controls are not supported for any content type:

- FramePositioningControl
- MIDIControl
- PitchControl
- RateControl
- TempoControl
- MetadataControl
- RecordControl is not supported for recording of video.

The video parameters width, height and encoding capture://video are not supported because video cannot be recorded.

HTTP streaming is not supported. The video and audio player implementations read all of the media data into a buffer before anything is played.

Mixing is not supported, that is, only one player can be playing at a time. For example, it is not possible to play a WAV file and a MIDI file simultaneously.

3.6 VideoControl Notes

If the video control's display mode is set to `USE_DIRECT_VIDEO`, the second parameter of the method `initDisplayMode` may be either a `Canvas` or the Nokia UI API's `Canvas` subclass '`FullCanvas`'.

In the method `setDisplaySize`, if the requested display size is different from the dimensions of the video clip, the image will be scaled to fit.

Method `setDisplayFullScreen(true)` is supported for both `Canvas` and `FullCanvas`; in `Canvas` it only sets the display to the entire `Canvas`, not the entire screen. The values returned by methods `getDisplayWidth` and `getDisplayHeight` are unaffected by this call (that is, they do not return the width and height of the full screen afterwards).

3.7 Time Bases

All players support the method `setTimeBase`, provided that the new Time Base was obtained from another player. They do not support `setTimeBase` with the MIDlets' own implementations of interface `TimeBase`.

3.8 RTP Streaming

The MMAPI specification defines locator syntax for RTP streaming. RTP streaming is not, however, supported from a locator, only from input stream.

4 Evaluate This Document

In order to improve the quality of documentation, we kindly ask you to fill in the [document survey](#).