



SavaJe JSR Implementation Notes

SavaJe JSR Implementation Notes

jsr_impl_2.5_v1.0_1

Published 09 May 2006 19:47
Copyright © 2006 SavaJe Technologies, Inc.

Legal information

SavaJe OS™ © 1999-2006, SavaJe Technologies, Inc. All rights reserved.

SavaJe OS v. 2.5 includes software and technology licensed from: ARM Ltd., Superscape Limited, Beatnik, Inc., EMBEDnet, Inc., HCC-Embedded Kft., XAudio Corporation (XAudio software © 1996-2001, XAudio Corporation. All rights reserved), Mezo, a Division of Cambridge Consultants, Ltd., Nexthaus, Inc., Openwave Systems, Inc., OZ Communications, Inc., PacketVideo Corporation, Sun Microsystems, Inc., and Tegic Communications, Inc. The Bluetooth® word mark and logos are owned by the Bluetooth SIG, Inc. and any use of such marks by SavaJe is under license.

SavaJe OS is a trademark of SavaJe Technologies, Inc. Sun, Java and all Java logos are trademarks of Sun Microsystems, Inc. All other product names are trademarks or registered trademarks of their respective owners.

This document is confidential and proprietary information of SavaJe Technologies, Inc. If disclosed at all, it is disclosed pursuant to a non-disclosure agreement between the recipient and SavaJe. This document is provided for informational purposes only, and SavaJe makes no warranties, either expressed or implied, in this document. Information in this document, including URL and other Internet web site references, is subject to change without notice. The entire risk of the use or the results of the use of this document remains with the user. Unless otherwise noted, any example companies, organizations, products, domain names, email addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of SavaJe Technologies.

SavaJe may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from SavaJe, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Table of Contents

1. Introduction	1
2. JSR-75 (PDA optional packages)	2
File access	2
PIM	2
Extended fields	2
3. JSR-82 (Bluetooth)	4
4. JSR-118 (MIDP 2.0)	5
TCK notes	5
5. JSR-120 (WMA 1.1)	7
6. JSR-135 (MMAPI 1.1)	8
Media controls	8
Player	10
Playback	11
7. JSR-139 (CLDC 1.1)	13
8. JSR-172 (Web Services)	14
9. JSR-184 (Mobile 3D Graphics API)	15
10. JSR-185 (Java technology for the wireless industry)	16
11. JSR-205 (Wireless messaging API 2.0)	17
12. JSR-209 (Advanced graphics for J2ME)	18

Chapter 1. Introduction

This document includes notes about SavaJe OS 2.5 support for JSRs. These notes are intended to assist developers in creating applications to be deployed on the SavaJe OS 2.5 platform.

SavaJe OS 2.5 can run applications in two modes: Connected Limited Device Configuration (CLDC) and Connected Device Configuration (CDC). Each mode supports only the API defined for that mode.

MIDlet suites run in CLDC mode (see *Developing MIDlet Suites for the SavaJe Platform*). For CLDC mode, SavaJe OS 2.5 supports all the JSRs designated as required or optional in Java Technology for the Wireless Industry (JSR-185), along with some additional JSRs related to the CLDC and MIDP environment.

MIDlet developers should refer to the JSR documentation for information on API usage. This document notes only areas where SavaJe OS support varies from what is described in the JSR API documentation. See <http://www.jcp.org/> for more information on specific JSRs.

Xlets run in CDC mode. For CDC mode, SavaJe OS 2.5 supports JSR-217 (Personal basis profile 1.1), JSR-218 (CDC 1.1), and JSR-219 (Foundation profile 1.1). In addition, SavaJe OS will support JSR-209 (Advanced graphics for J2ME), which is not yet in final status. Xlets can also use other JSRs specified here, with the exception of JSR-118 (MIDP 2.0). Xlets also have access to SavaJe services APIs (`net.savaJe.*` classes).

Chapter 2. JSR-75 (PDA optional packages)

SavaJe OS 2.5 provides virtually complete support for JSR-75 file access (with the limitations noted below) and partial support for JSR-75 PIM. Complete support for the JSR-75 File and PIM APIs is planned for SavaJe OS 2.5.

File access

The file systems to which an application has file access, and the level of access (read, write, read-write) are constrained by the security domain in which the application is running. Consult the operator for the constraints that apply to the security domain in which your MIDlet suite will run.

Untrusted MIDlets running on SavaJe OS have read and write access to the following directories:

```
/ramfs/tmp      (in-memory file system)
/cf0            (mini-SD card)
```

PIM

The SavaJe information and PIM data service maintain a persistent PIM database in the flash file system.

SavaJe OS 2.5 does not support import/export (`fromSerialFormat()`/`toSerialFormat()`). Accordingly, there is no support for the vCard 2.1 and vCalendar 1.0 data formats, or the Quoted-Printable and BASE64 encoding formats.

SavaJe OS 2.5 provides a security model with access rights for read only, write only, and read-write. For more information on MIDP permissions, see *Developing MIDlet Suites for the SavaJe Platform*.

Contact implements the PIMItem interface. The PIMList that a PIMItem belongs to determines what fields a PIMItem can support. Call the `PIMList.getSupportedFields()` method to determine the fields supported within a particular PIMItem.

Extended fields

SavaJe OS 2.5 provides several extended fields relating to PIM access. These fields are part of interfaces in the `net.savaJe.me.pim` package that extend the corresponding interface in `javax.microedition.pim`.

For example, extended fields for a Contact are in `net.savaJe.me.pim.Contact`. If an application uses an extended field for a Contact, it should import `net.savaJe.me.pim.Contact` and use it as a `javax.microedition.pim.Contact`. Since the fields are `final` constants, the `net.savaJe.me.pim` class does not appear in the byte code, but the field values will appear explicitly wherever they are used.

Note that to ensure platform-independent behavior, the use of any extended field should be guarded by a call to `PIMList.isSupportedField()`.

For documentation, see the `net.savaJe.me.pim` Javadoc.

PIM extended fields (`net.savaJe.me.pim.*`)

- - Contact.ATTR_VOICE
 - Contact.IMID
 - Contact.IM_SERVICE
 - Contact.MAILER
 - Contact.PRESENCE
 - Contact.PRESENCE_STATE
 - Contact.RINGTONE
 - Contact.VOICE_TAG

- - Event.ALARM_SOUND
 - Event.ALARM_TEXT
 - Event.ALARM_VIBRATE

- - PIM.PACKAGE_DISCOVERY_KEY

- - PIMItem.INFO_SVC_URI
 - PIMItem.IRMC_LUID
 - PIMItem.MAX_COMMON_EXTENDED_ATTRIBUTE
 - PIMItem.MAX_COMMON_EXTENDED_FIELD

- - RepeatRule.DEFAULT_COUNT
 - RepeatRule.MAXCOUNT
 - RepeatRule.MONTHS
 - RepeatRule.WEEKDAYS
 - RepeatRule.WEEKS
 - RepeatRule.EXCEPTION_DATE_LABEL
 - RepeatRule.RULE_LABEL
 - RepeatRule.WEEKDAY_STRING

- - ToDo.ALARM
 - ToDo.ALARM_SOUND
 - ToDo.ALARM_TEXT
 - ToDo.ALARM_VIBRATE
 - ToDo.STATUS

Chapter 3. JSR-82 (Bluetooth)

SavaJe OS 2.5 supports the `javax.bluetooth` APIs specified by JSR-82. The `javax.bluetooth` APIs include support for device discovery, service discovery, L2CAP (logical link control and adaptation protocol), and RFCOMM protocol.

Support for the `javax.obex` portion of JSR-82 will be available in a future release.

SavaJe OS supports the following Bluetooth profiles:

- Handsfree
- Headset
- Dial-up networking
- File transfer
- Object push
- Synchronization

Chapter 4. JSR-118 (MIDP 2.0)

SavaJe OS 2.5 supports all mandated (MUST) and recommended (SHOULD) items in JSR-118 with the exceptions and conditions noted below.

For supported audio and video mime types, see [JSR-135 implementation notes](#) [p. 8].

A feature of MIDP 2.0 and WMA 2.0 (JSR-205) is the ability to launch a MIDlet as a result of a message arriving at the hosting device. Messages that conform to MIDP Push Registry entries specified in the Java Application Descriptors (JAD) of MIDlets can cause the MIDlets to be launched. MIDlets can also register to be launched based on the same or similar criteria through the use of the MIDP Push Registry API. The Message Service and MIDP Application Management Service (AMS) implementations facilitate this feature. See JSR-118 and JSR-205, appendix E for specification details. SavaJe OS 2.5 supports the SMS protocol for PushRegistry messages.

PushRegistry ServerSocketConnection is implemented, but disabled by carrier request for security reasons.

You can use the MIDlet `platformRequest()` method to initiate a voice call, to initiate the installation or update of a MIDlet suite, and to use the system WAP browser to view a web page. The SavaJe platform uses its content handler service to respond to platform requests.

SavaJe OS 2.5 supports use of the `CommConnection` interface to access available RS232 serial ports (if any) on the handset.

If a MIDlet requests an unsupported baudrate with `CommConnection setBaudrate()`, the system substitutes a valid baudrate, which may be discovered with the `getBaudRate()` method.

The SavaJe application manager uses an HTTP POST to report the status of an installation, update, or removal to the URL specified in the `MIDlet-Install-Notify` or `MIDlet-Delete-Notify` attribute. The URL protocol must be `http`. If the SavaJe application manager is unable to send the status report or does not receive a server reply, the application manager resends the status report. If this operation fails again or is not followed by a reply from the server, the application manager does not resend the status report once the installation, update, or removal process has been completed.

SavaJe OS 2.5 supports the use of X.509 certificates for signed MIDlet suites.

The SavaJe OS 2.5 implementation of `javax.microedition.lcdui.TextField` does not provide a UI for copying a phone number from the user's phone book to a `TextField` with the `TextField.PHONENUMBER` constraint when that `TextField` has input focus.

RMS data is stored in the flash file system.

Up to 32 audio streams are supported.

A player cannot use another player's `TimeBase`.

TCK notes

The following implementation notes apply to SavaJe OS results on the MIDP `javax.microedition.lcdui` TCK tests. Unless otherwise noted, these do not represent TCK test failures, but they may be useful to MIDlet developers.

- `AlertType`

- For `playSound()`, only the `ALARM` sound plays. This is permitted by the specification.
- `Canvas`
 - For `getKeyName()`, note that SavaJe OS currently has no `CLEAR` key; the End key is Escape. This is permitted by the specification.
 - SavaJe OS does not support `keyRepeated` events. This is an optional feature.
 - SavaJe OS does not support `pointerDragged` events. This is not currently applicable since none of the platforms we support have a mouse-like input device.
- `CustomItem`
 - SavaJe OS does not support `keyRepeated` events. This is an optional feature.
 - SavaJe OS does not support `pointerDragged`, `pointerPressed`, or `pointerReleased` events. This is not currently applicable since none of the platforms we support have a mouse-like input device.
- `Display`
 - The SavaJe implementation of vibration limits the vibration to 2 seconds due to hardware limitations. This is permitted by the specification.
- `Font`
 - The font style `STYLE_UNDERLINED` breaks the underline between words. This is permitted by the specification.
- `Form`
 - The layout of items by SavaJe OS may be different than what the MIDlet toolkit (based on Sun's Wireless Toolkit or WTK) does. `TextFields` are laid out on successive lines. `CustomItems` are sized to the text label's width.
- `List`
 - An empty `List` displays a 4x4 pixel dot in the upper left-hand corner because the empty list gets focus and displays a focus border. This is an allowed look-and-feel variation.

Chapter 5. JSR-120 (WMA 1.1)

SavaJe OS 2.5 supports all requirements of the Wireless Messaging API 1.1 (JSR-120). Note that SavaJe OS 2.5 also supports the Wireless Messaging API 2.0 (JSR-205). See that section for the latest information on wireless messaging support.

Preliminary

Chapter 6. JSR-135 (MMAPI 1.1)

With the exceptions noted below, SavaJe OS 2.5 supports all mandated (MUST) and recommended (SHOULD) items in JSR-135.

Media controls

SavaJe OS 2.5 supports media controls as indicated below:

Media	Implemented	Not implemented
Image	GUIControl, VideoControl	
SMIL	MetaDataControl, GUIControl, VideoControl, PresentationControl. Will also provide controls provided by plugins for the media specified within the SMIL document.	
Sampled Audio	VolumeControl, MetaDataControl	StopTimeControl, PitchControl
MIDI	VolumeControl, MetaDataControl	MIDIControl, TempoControl, PitchControl, StopTimeControl
Tone Sequence	ToneControl, VolumeControl	StopTimeControl
Interactive MIDI	Not supported	
Video	VideoControl -- setDisplayFullScreen() supported for images only;	FramePositioningControl, StopTimeControl
	getSnapshot() only supported for camera .png or .jpeg image (JSR-135 scenario 11)	
Video Capture	RecordControl, VideoControl, VolumeControl	
Still Image Capture	VideoControl	

Other Controls

- GUIControl: supported.
- MetaDataControl: supported for MIDI files and Sampled Audio only; available keys are AUTHOR_KEY, COPYRIGHT_KEY, DATE_KEY, and TITLE_KEY
- RateControl: not supported.
- RecordControl: supported for capable devices.

SavaJe OS 2.5 does not support multi-player synchronization.

To determine programmatically what content types are supported, applications should call `javax.microedition.media.Manager.getSupportedContentTypes()`.

SavaJe OS 2.5 supports the following audio and video mime types:

Mime type	File format	Data encoding
Audio		
audio/3gpp (RFC 3839)	3GP	AAC-LC, AMR-NB
audio/amr	AMR	AMR-NB (speech)
audio/midi	MIDI	MIDI -- type 1 & 0 files; conforms to Scalable Polyphone MIDI Specification
audio/sp-midi	SP-MIDI	SP-MIDI -- type 1 & 0 files; conforms to SP-MIDI Device 5-24 Note Profile for 3GPP
audio/mp4a-latm	mp4	AAC-LC (48kHz max)
audio/mpeg	mp3	MP3
audio/x-tone-seq	jts	defined by JSR-135
audio/x-wav	wav	PCM 8bit, PCM 16bit mono/stereo
Video (with or without audio)		
video/3gpp	3gp	audio data -- AAC-LC, AMR-NB video data -- MPEG4 Visual Simple Profiles Level0; H.263 Profile 0 Level 0
video/mp4v-es	mp4	audio data -- AAC-LC (no AMR-NB) video data -- MPEG4 Visual Simple Profiles Level 0; H.263 Profile 0 Level 10
video/x-msvbidel	avi	audio data -- MP3, aLaw, uLaw, PCM video data -- bitmap video
video/h263-2000	h263	RTP streaming video payload; H.263 Profile 0 Level 10; H.263 Profile 3 Level 10
Image		
image/gif	gif	GIF87a, GIF89a
image/jpeg	jpeg	JPEG
image/png	png	PNG with alpha channel
image/vnd.wap.wbmp	wbmp	WBMP

Notes on audio playback limitations

The following notes apply for audio playback with non-WAV file formats:

- Content length must be known and the content must be seekable (e.g., as when playing from a file).
- Playback with non-markable InputStreams is limited to 1MB. The entire InputStream will be read upon player creation to calculate content length.
- For playback with InputStreams that are markable, the data will be read multiple times. Once, immediately upon player creation to calculate content length and subsequently during playback.
- Playback over HTTP/HTTPS URLs will be limited to 1MB per player. If the server does not provide content length, the entire stream will be read at player creation.
- Playback using DataSource is limited to SourceStream objects that are RANDOM_SEEKABLE and that provide a valid content length.

Player

Player `setMediaTime()` and `setTimeBase()` are not supported (throw a `MediaException`); `getTimeBase()` is supported.

Support for the following Player APIs is media dependent.

API	Media
<code>getDuration()</code>	MediaPlayer -- supported sample audio/video media types
	MIDI and ToneSequence -- returns sequencer value
	Image/AudioCapture/Camera -- either a snapshot or live; returns <code>Player.TIME_UNKNOWN</code>
<code>getMediaTime()</code>	MediaPlayer -- supported sample audio/video media types
	MIDI and ToneSequence -- not supported; returns <code>Player.TIME_UNKNOWN</code>
	Image/Audio/Capture/Camera -- not supported; returns <code>Player.TIME_UNKNOWN</code>
<code>setLoopCount()</code>	MediaPlayer -- supported for supported sample audio/video media types
	MIDI and ToneSequence -- silently ignores unless improper state
	Image/AudioCapture/Camera -- silently ignores unless improper state

SavaJe OS 2.5 supports Player event types as follows:

PlayerListener field	Supported
<code>BUFFERING_STARTED</code>	No

PlayerListener field	Supported
BUFFERING_STOPPED	No
CLOSED	Yes
DEVICE_AVAILABLE	No
DEVICE_UNAVAILABLE	No
DURATION_UPDATED	Yes
END_OF_MEDIA	Yes
ERROR	Yes
RECORD_ERROR	Yes
RECORD_STARTED	Yes
RECORD_STOPPED	Yes
SIZE_CHANGED	No
STARTED	Yes
STOPPED	Yes
STOPPED_AT_TIME	Yes
VOLUME_CHANGED	Yes

Playback

The SavaJe JSR-135 implementation allows for creating media players with the following sources:

- **File Locator:** Playing a file sitting either on an SD card or on the FFS using a URI of the form `file://`. MIDlets with the appropriate file access permissions can use this.
- **InputStream:** A fairly simple Java data structure that supports read/skip. InputStreams do not necessarily have the notion of seek. Seek capability is implemented by exploiting a feature of InputStreams called "Markable." Another shortcoming of InputStream objects is that there is no means of querying the length of the content being played. InputStreams are used quite extensively by games and other popular MIDlets.
- **DataSource:** A J2ME data structure that provides more capability than a simple InputStream with the additional methods to support seeking. DataSource also allows for querying the length of the content being played.
- **URI:** Creating a player that reads content from a given URL and plays it. Supported URIs are `http://` and `https://` URIs such as `resource://` are not supported.

Playback limitations

File Locators

This use case has no known limitations on it.

InputStream

If the InputStream object supplied for playback is not markable, then the SavaJe/OS JSR-135 implementation internally wraps the given InputStream into a BufferedInputStream thereby gaining seek capability. If the InputStream object is markable, then no additional wrapping is necessary.

Requests from the multimedia engine for data from arbitrary positions are serviced by navigating the `InputStream`. For example, if the stream has reached a position of 23K and the engine needs to read some data at position 10K, the implementation will then first reset the `InputStream` position to 0, invoke `InputStream.skip()` for the first 10K bytes and then provide the data. For non-markable `InputStream` objects, the `mark()/reset()/skip()` requests will be shielded by the `BufferedInputStream`. However, a `MIDlet` that provides a markable `InputStream`, will directly see these requests.

The `MIDlet` should be prepared that the data from its `InputStream` object will be read, perhaps multiple times, via `mark/reset` as described above.

As discussed earlier, all non-WAV playback requires that the content length be known before playback can begin. Since `InputStream` objects have no means of providing content length, length is calculated by reading through the entire `InputStream` when the `Player` is created. If a `MIDlet` provides a non-markable `InputStream` object, then the system's implementation of `BufferedInputStream` effectively buffers the entire content of the `InputStream` into memory when the player is first created. Successful playback therefore depends on the length of the content being played and the amount of free system memory.

DataSource

This object provides the JSR-135 implementation of two advantages over an `InputStream` object (a) The ability content (b) The ability to seek. For the SavaJe JSR-135 implementation, the `DataSource`'s `SourceStream` implementation must be `RANDOM_SEEKABLE` and the `getContentLength()` method must return a valid length.

HTTP and HTTPS

For these URIs, a `BufferedInputStream` is created that encapsulates the corresponding HTTP/HTTPS connection. When the player is instantiated, an attempt is made to calculate the content length via the connection. Most modern web servers provide a valid content length.

If an HTTP connection does not provide content length, then as with input streams, the entire stream is read (i.e., buffered), in an attempt to calculate content length. Such a connection will therefore see its entire media file being read upon player startup before even the first second of playback.

A HTTP connection providing content length will see a sequential reads to the content as playback progresses.

As with `InputStream`, the amount of content that can be played over HTTP/HTTPS is limited to 1MB.

Chapter 7. JSR-139 (CLDC 1.1)

Running in CLDC mode, SavaJe OS 2.5 meets all the requirements of the Connected Limited Device Configuration 1.1.

Preliminary

Chapter 8. JSR-172 (Web Services)

SavaJe OS 2.5 supports XML parsing: `javax.xml.parsers`, `org.xml.sax`, `org.xml.sax.helpers`. SavaJe does not support the XML Remote Procedure Calling protocol (XML-RPC).

Preliminary

Chapter 9. JSR-184 (Mobile 3D Graphics API)

SavaJe OS 2.5 provides complete support for JSR-184.

Preliminary

Chapter 10. JSR-185 (Java technology for the wireless industry)

JSR-185 is an umbrella JSR that incorporates JSR-030 (Connected Limited Device Configuration 1.0), JSR-118 (MIDP 2.0), JSR-120 (Wireless Messaging API 1.1), and JSR-135 (Mobile Media API 1.1). Implementation notes for these JSRs are provided in the appropriate sections of this document.

Preliminary

Chapter 11. JSR-205 (Wireless messaging API 2.0)

WMA 2.0 (JSR-205) is a superset of WMA 1.0 (JSR-120). The following describes SavaJe OS 2.5 support for JSR-205.

Supported

- Appendix A -- GSM SMS Adapter
- Appendix B -- GSM Cell Broadcast Adapter
- Appendix D -- MMS Adapter
- Appendix E -- Deploying JSR-205 interfaces on a MIDP 2.0 platform

Not supported

- Appendix C -- CDMA IS-637 Adapter

Chapter 12. JSR-209 (Advanced graphics for J2ME)

JSR-209 is currently in the final draft stage. The API in the current SavaJe implementation may differ from the final version. This is an early access release, and there are several areas where implementation is incomplete. In addition, because JSR-209 components have the same names as components in J2SE, you may expect the behavior to be identical. This is not always the case; consult the API documentation for the expected behavior.

The following are known issues with the current JSR-209 implementation:

- Static `main()` with `Frame` not supported:

SavaJe currently does not support Graphical applications that rely on a static `main()` and creating a `Frame`. Applications must use the `Xlet` model and get the root container from the `XletContext.getRootContainer()` method.

- Arbitrary shape clipping is not currently supported.
- Images are currently painted with an arbitrary transformation matrix, not an `AffineTransformOp`.
- Support for image types other than `BufferedImage` image type is not yet implemented.

SavaJe Technologies

100 Apollo Drive
Chelmsford ■ MA 01824

tel 978 ■ 256 ■ 6521
fax 978 ■ 256 ■ 8386

