Implementation Guidelines

JTWI

Version: 2.0.0

Owner: Hakim Mendjeli & Osamu Otaka
Authors: Osamu Otaka

Table of Contents

1 INTRODUCTION 4

2 PLATFORM ARCHITECTURE OVERVIEW 5

3 PREFERENTIAL BEHAVIOUR FOR INCOMING CALL 6
3.1 Java Application behavior: incoming call event. 6
3.2 User notification within the status area 6
3.3 User Operation While Incoming Call Event 7

4 CONFLICTING BEHAVIOUR OF NATIVE FUNCTION DURING JAVA EXECUTION 8
4.1 Screen lighting control 8
4.2 Sound volume control 8

5 OFF-SCREEN 9

6 GIF SUPPORT 10

7 ALPHA BLENDING 11

8 PLATFORM REQUEST 12

9 JSR135 ADDITIONAL IMPLEMENTATION 13
9.1 Camera 13
9.1.1 Finder 13
9.1.2 Capture sound 13
9.1.3 Player 13

10 MIDXLET-SCREENSIZE 15

11 REFERENCES 18

Table of Figures

Figure 1 Platform Architecture Overview 5
Figure 2 PlatformRequest Sequence 12
Figure 3 Java Application Finder Display 13
Figure 4 Finder Display on Java Application 13

1 Introduction

As part of the Java Service to be launched by Vodafone Group, “VFX terminals” will be required to implement the JTWI specification.
This document describes the features and behaviour of a handset related to JTWI implementation, as a supplement to the TCD.

2 Platform Architecture Overview

The Java VM for JTWI is described based on the assumption that the JTWI Platform can be configured as shown in Figure 1 below;

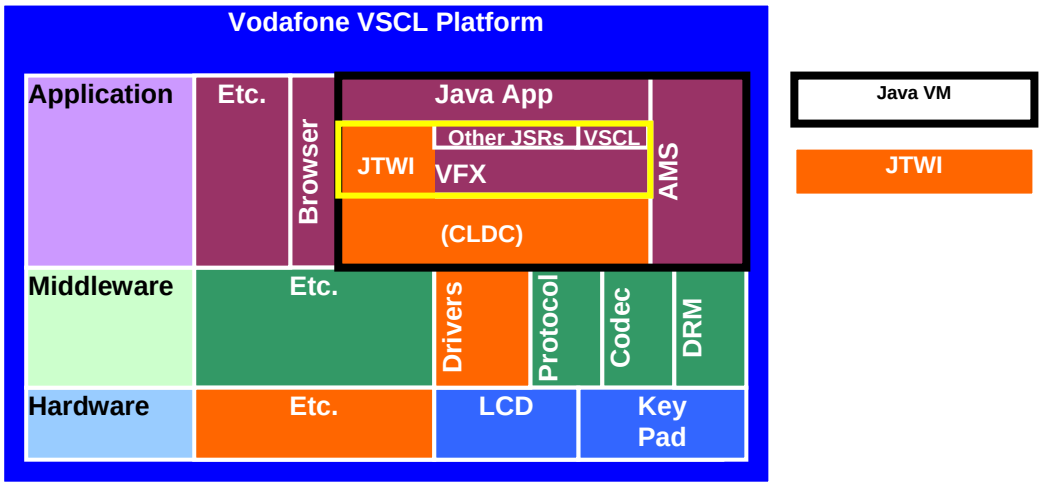


Figure 1 Platform Architecture Overview

The table below classifies and describes individual components and their functions.

Item	Description
AMS	AMS is an abbreviation of Application Management Software. This is Software which verifies Java Applications/controls execution.
CLDC	CLDC is an abbreviation of Connected Limited Device Configuration, which includes Virtual Machine (KVM) as well. The CLDC in this document refers to the CLDC1.1.
VFX	VFX stands for “Vodafone effects”, (“FX”, sounds like “effects”). It is also an abbreviation of “Vodafone VFX”. VFX is a component of a variety of the Java Profiles defined by Vodafone.
VSCL	Vodafone Specific Class Library2.0/2.1. VSCL is a proprietary API which Vodafone specified, and is evolved from JSCL.
JTWI	Java Technology for the Wireless Industry. This document includes MIDP2.0,CLDC1.1,JSR120,JSR135.

3 Preferential behaviour for incoming call

This section describes the handset's behaviour during incoming events while a Java application is running

3.1 Java Application behavior: incoming call event.

The following settings define the handset's behaviour when a Java application is running during an incoming call event.

Table 1 Incoming priority operation setting

Setting	Description
Incoming Call Priority Operation	If a native event, such as incoming call or a received message, while a Java application is running, the following operations should occur: - The Java application is paused - The notified event is handled appropriately. In this case, incoming call event shall not be notified to Java VM.
Incoming Call Notification Indication	If a native event occurs as a result of incoming call or received mail, while a Java application is running, the Java application shall continue to be run, and the event shall be notified to the Java application. If an incoming event occurs, the user should be notified in the status area.

3.2 User notification within the status area

The following table describes the notifications of incoming events while a Java application is running. The notification message should be displayed in the status area.

Table 2 Description of the possible user notifications within the status area:

Event			Description
Voice	Incoming voice calls		Incoming call notification, caller name, calling number
	TV Call		Incoming call notification, caller name, calling number
SMS	SMS	Receiving SMS	Incoming call notification, mail address or sender name
		Receiving delivery confirmation	Incoming call notification,
WAP Push	MMS	Receiving Notification (Auto receiving OFF/MMS)	Incoming call notification, mail address or sender name
		Receiving Notification (Auto receiving ON/MMS)	Incoming call notification, mail address or sender name
		Receiving Notification (Auto receiving OFF/E-mail)	Incoming call notification, mail address or sender name
		Receiving Notification (Auto receiving ON/E-Mail)	Incoming call notification, mail address or sender name
		Receiving delivery confirmation (Notification)	Incoming call notification
		Receiving delivery confirmation (Retrieval)	Incoming call notification
		Memory full	Notify memory as full
	Service Indication	HIGH	-
		MIDDLE	-
		LOW	-
	Service Load		-
	Cache Operation		-
	DRM Rights		Incoming call notification
	Location		Incoming call notification
	Provisioning(bootstrap)		-
	Server Alert Sync		Incoming Notification
CBS	CBS		Incoming call notification
External	Bluetooth		-
	USB、Serial (Receiving Uplink)		-
Alarm	Native alarm		Alarm notification, comment
	Low battery		-
Sync	Automatic Synchronization		-

* Server Alert Sync and Automatic Synchronization are applicable if background conflict process cannot be done.

Table 3 Timing of the notification display within the status area

Event	Time
Incoming voice calls	Immediately notifies the user when the call first comes in, and continues to notify the user during the call.
Incoming event other than voice	Native implementation dependent

3.3 User Operation While Incoming Call Event

When the notification of an incoming call is displayed in the status area, the following actions are assigned to the handset's keypad:

Table 4 Handset's behaviour in response to user's key press

User operation	Application in execution
Accept key	Pause Java application and conduct incoming operation with Native.
Reject key	Continue Java application and hang up the call.
Any key(numeric keypad, *, #)	Continue Java application execution.

4 Conflicting behaviour of Native function during Java execution

This section describes how the handset should handle conflicting behaviors between handset functions, Java environment and Java applications.

4.1 Screen lighting control

The terminal shall follow the behaviors described below in regards to backlight function when a midlet is running.

Table 5 Backlight setting

Setting	Description
Always ON	Light ON always
Always OFF	Light OFF always
Link to Backlight	Interlocked with key operation

Note that the Java application may toggle the ON/OFF status dynamically (i.e. within a java midlet, as defined in MIDP2.0), which will have the same effect as setting “Always ON”, or “Always OFF”.

4.2 Sound volume control

Volume setting function is required when executing Java application. The handset must playback the sound according to these settings when a midlet is running.

Table 6 Java Sound volume setting

Setting	Description
Sound ON(1,2,3,4,5)	Sound is played with the set-up volume.
Sound OFF(0)	Even if application plays sound, Volume is always set to 0.

Note that when the Java application programmatically changes the volume of a sound it is playing, the volume change shall be applied as a factor to the sound volume set by the user. E.g. if the user selects sound ON 3 (which is 3/5 volume) and the program sets 75% volume, then the actual volume of the sound played by the program shall be 75% of 3/5 = 9/20 of maximum volume.

When silent mode is activated, it overrides any other sound setting.”

Table 7 Sound control when silent mode is set

Setting	Description
Silent mode setting OFF	Comply with Java sound volume control setting.
Silent mode setting ON	Comply with the sound volume of silent setting. (Set as minimum sound volume if step method is set)

5 Off-Screen

The handset Off-Screen specification is defined as follows.

Table 8 Minimum number of Off-Screen Image Objects supported

Index	Value
OffScreens Image Objects	512 or more.

6 GIF support

GIF87a, GIF89a (for Animated GIF, the first frame still image shall be displayed) shall be supported in LCDUI Image.createImage

7 Alpha Blending

Image semitransparent effect (alpha blending) shall be supported
Handset shall be supporting 256 alpha levels at least.

Reference to: TCD JAVA "TCD-GJAV-MUL-001134"
Reference to: VFX Spec "section 4.1.1.2"

8 Platform Request

It shall be possible to activate the browser and originate a voice call with MIDlet.platformRequest as follows:

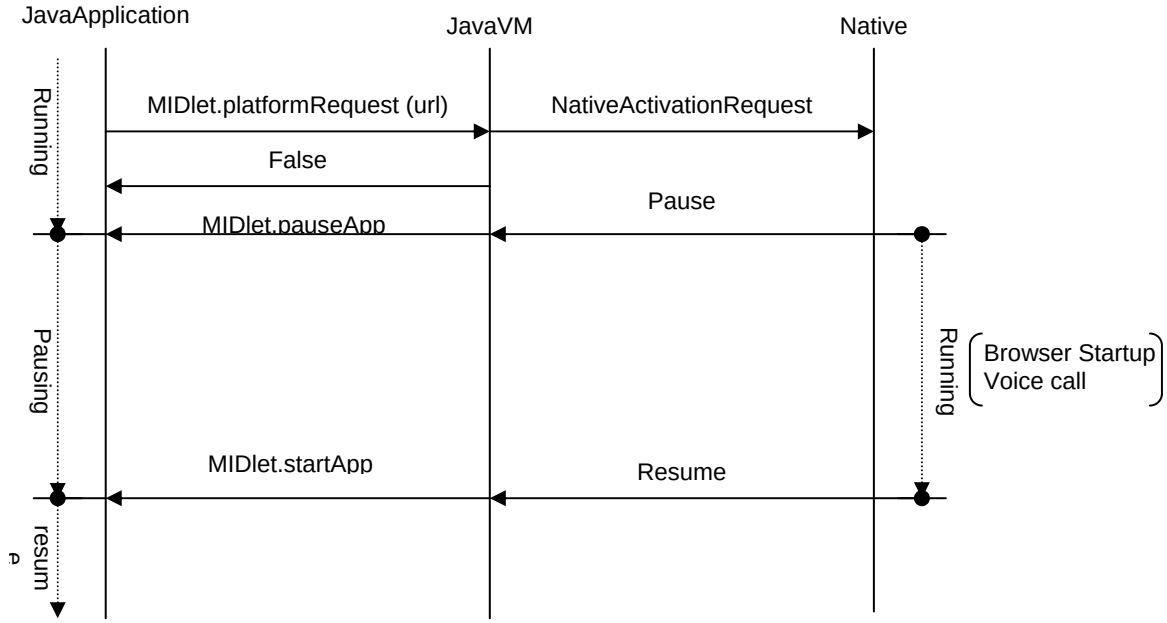


Figure 2 PlatformRequest Sequence

When platform application (browser or voice call) is terminated, the handset shall resume Java application.

・ Confirmation Screen Display

In case of activation of Native application due to Platform Request, confirmation screen shall be displayed in Native. In addition, the native application (browser or voice call) shall not be activated if the user rejects it.

Reference to: VFX Spec "section 4.1.1.9"

9 JSR135 additional implementation

Written below describes the conformance of MMAPI.

9.1 Camera

JSR135 specifies camera support. The following clarifies this:

9.1.1 Finder

Finder on Java application shall display the center as same as center of Native finder.

If a finder on Java application is smaller than Native finder size, then the native finder size shall be centered in the Java finder, and clipped down to the size of the Java finder. (See figure below)

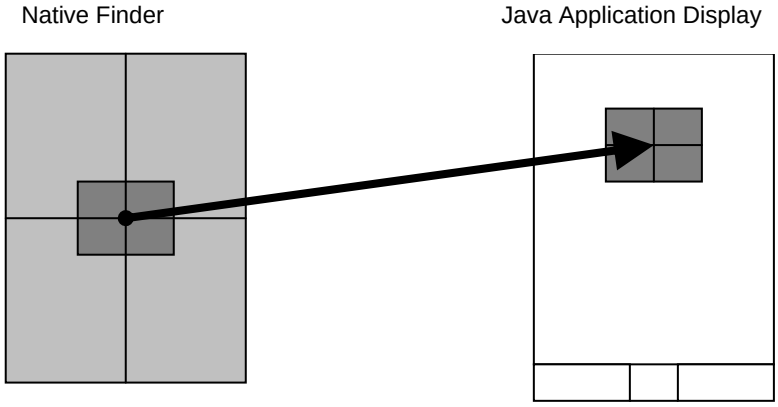


Figure 3 Java Application Finder Display

If a finder on Java application is larger than Native finder size, then the extra area around the native finder will be displayed whited out. (See figure below).

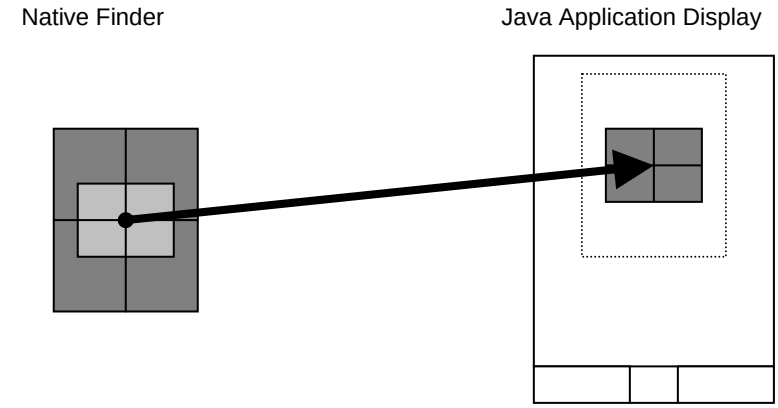


Figure 4 Finder Display on Java Application

9.1.2 Capture sound

The sound of Native shutter sound for Java capture process shall be same as Native configuration.

9.1.3 Player

Parameter of Player creation for Camera shall be able to be configured with ”width” and “height”, but shall ignore ”video_enc” and ”fps”.
If the “width” or “height” exceeds the Native maximum size, then throw an Exception.
If no parameters are set, use Native maximum size for “width” and “height”.

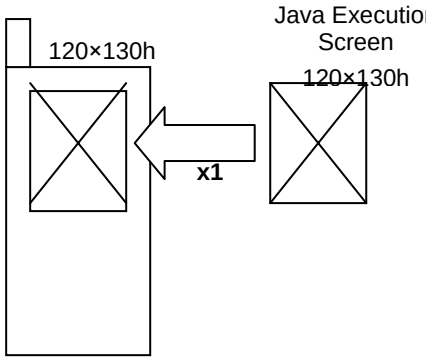
Reference to: VFX Spec “section 6.2.7”

10 MIDxlet-ScreenSize

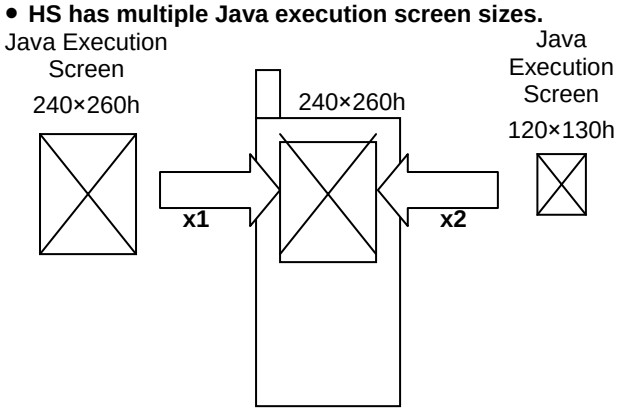
Definition

Usually, since a handset has only one Java execution screen size, the size is fixed. This specification describes a mechanism for effectively mapping a virtual screen size to the physical screen, if requested by the JAD paramemter MIDxlet-Screen-Size.

- **HS has only one Java execution screen size.**



- **HS has multiple Java execution screen sizes.**



A Jad file may specify the following attributes:

Attribute Name	Attribute Value
MIDxlet-ScreenSize	W,H W：Width targeted by application H：Height targeted by application *Numbers are all single byte. *Unit: dot *MAX 32 bytes

Description of the mechanism for determining how to process attributes

The attribute value could be defined in the following way:

- Specify fixed value
- Default value
- Jad error

(1)Specify Fixed Value

The execution screen size targeted by Java application is described using a fixed width and height:

MIDxlet-ScreenSize:W,H

The handset shall determine compatibility as so:

If the W x H specified in the JAD file is larder than the physical screen size (I.e.W > physical width, or H > physical height) then the AMS shall reject the JAD file. Otherwise, proceed to retrieve the JAR. When starting the application, W<= physical width, H <= physical height; otherwise the application is incompatible.

The handset may support multiple increments of virtual screen size mappings. E.g. if the physical screen size is 260x320; that phone may implement 3 virtual screen size mappings: 260x320 (no transformation needed), 130x160 (scale up by 2 … each “pixel” drawn by the application occupies 4 physical pixels on the screen), 173x213 (scale up by 1.5).

The most appropriate virtual screen mapping size shall be selected for the application. If there are multiple virtual screen mappings implemented in the handset, the mapping that results in the maximum display magnification shall be selected.

The screen size reported to the MIDlet (via Canvas.getWidth and Canvas.getHeight) shall be the same size as specified in the MIDxlet-Screen-Size parameter (unless it is 0,0 – in which case the physical screen size shall be reported.)

Ex 1 Handset A, B, or C with the following virtual screen size mappings running an application with MIDlet-ScreenSize:160,160

Handset A 240×240(x1)、120×120(x2)

Handset B 240×240(x1)、160×160(x1.5)

Handset C 180×180(x1)

*() = Display magnification

For Handset A, the compatible Java virtual screen size is 240×240. So, this handset shall display the application at 240×240. For Handset B, the compatible sizes are 240×240 and 160×160. So, 160×160 will be selected, and the application will be scaled up by 1.5 as the virtual screen size with the maximum display magnification shall be adopted for display. For Handset C, since there is only one execution screen size, 180×180 shall be adopted for display.

(2)Default Value

When an attribute is not specified, the following shall be assumed as the default value.

MIDxlet-ScreenSize: 0,0

When this attribute is specified, the virtual screen size with the highest display magnification shall be selected without any conditions.

(3)Jad Error

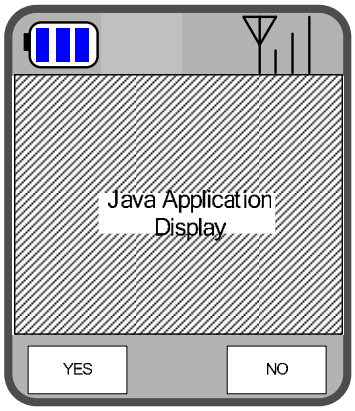
When the following conditions occur, notify Java Runtime Environment the Jad format error.

- No comma, or multiple commas
- Either numeric value or range value is missing before or after a comma.
- There is a blank space in the attribute value. However, ignore the blank space between attribute and colon.
- Numeric value or range value is not single-byte.
- Numeric value or range value is not integral.
- The attribute value exceeds 32 bytes.
- The maximum Java virtual screen size in a handset is smaller than the attribute value (even if just a little.)

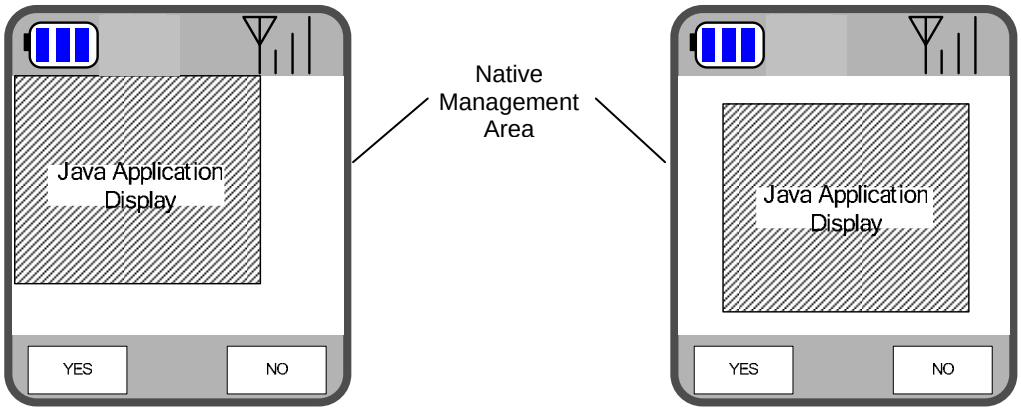
Display Operation

The following examples show how special attention must be taken for a handset to display the Java virtual screen.

- Handsets only need to display the Java virtual screen as described above and do not need particular processing for the screen (the drawing within the range is an application matter).



- When displaying the selected Java virtual screen (including zooming at equal magnification between width and height), the Java display screen shall not be placed with a disproportionate emphasis on one side of the handset screen due to some space on the other side of the handset screen (see the left picture of the figure below), the Java display screen shall be centralized (see the right picture of the figure below). Drawing in the area outside of the java display screen is not allowed. Note that the "java display screen size" is the area of display of the MIDlet's MIDxlet-Screen-Size after any magnification by virtual screen mapping.



Reference to: VFX Spec "Appendix E"

11 References

Reference	Description
TCD JAVA	"Vodafone Terminal Capability Definition Game & Java"
VFX Spec	"Vodafone VFX Specification"
JTWI	“Java™ Technology for the Wireless Industry”