

# Emulator Configuration and Customization

## 1 Introduction

This Document describes the MidletRunner configuration, startup process and skin customization.

## 2 Emulator and Midlet Startup Process

The emulator startup process differs slightly, depending on if it is used as applet or from the command line. The lookup sequence for properties is always as follows:

1. Current set of properties (including properties from the jad file or meta-inf)
2. Fall back to System.properties for the command line version and applet properties for the emulator applet.

The following sections describe both variants of the startup process.

### 2.1 Start from the Command Line

The initial set of emulator properties is determined from command line parameters. Command line parameters starting with “-” are treated as property names, the following argument is treated as property value.

An independent parameter without the “-” prefix determines the MIDlet main class or JAD file. If the parameter ends with “.jad”, it will be treated as JAD file name. Otherwise, it will be treated as MIDlet class name, implicitly constructing a JAD file with a corresponding “MIDlet-1” entry.

Parameters prefixed with “JAD:” are copied to the JAD file.

### 2.2 Emulator Applet Start

When the emulator is started as applet, there is no explicit initial set of properties. Instead, the applet parameters are taken into account by the property fall back mechanism described at the beginning of this section, making all applet parameters available as properties (until overwritten by a later step of initialization).

### 2.3 Common Initialization and Localization

Before starting the MIDlet, the Application Manager performs a set of general system setup steps:

1. If the “microedition.locale” property is not set (including applet parameters, see above), set it to the system default locale.
2. Run platform specific implementation initializers (org.me4se.psi.\*)

3. Locale-specific properties are loaded from the resource “locale-<locale>.properties”. If country specific properties are not found, language specific locales are searched. Example: “locale-de-DE.properties”, “locale-de.properties”.
4. If the “skin” property is set, load the device skin properties from the corresponding resource and set up the skin accordingly.

If a JAD file has already been set by command line options, the JAD file is used to start up the MIDlet. Please note that no attempt is made to load the JAR file – the MIDlet JAR file must be part of the classpath already.

If no JAD file is available yet (always in the Applet case), the JAD file is determined by the following steps:

1. Use the property value “jad” as URL for the JAD file
2. Use the resource “/me4se.properties” as JAD file.
3. Search for a /META-INF/MANIFEST.MF resource containing a MIDlet-1 property

The first JAD file found will be used to start the MIDlet.

### **3 Emulator Skins**

Each emulator skin zip file contains a top level file named “skin.properties”. Edit this file to customize the emulator skin.

Please note that all referenced files need to be contained in the customized emulator skin zip file.

## 4 Display Properties

Property	Value / Type	Description
default_image pressed_buttons_image highlighted_image	Image file name	Image files showing the device in neutral mode and with the buttons pressed and highlighting
icon.<name>	x, y, initial state	Defines the position and initial state of an indicator icon. Valid names are “up” and “down”; valid states are “on” and “off”
icon.<name>.<state>	Image file name	Defines the image for the given icon state. (icon.up.on, icon.up.off, icon.down.on, icon.down.off)
colorCount	Integer	number of colors / grayscales available
isColor	Boolean	Determines whether the device display is color or grayscale
scale	Integer	Magnification factor
screenBackgroundColor	RGB hex int (e.g. 0xb6b6aa)	The background color of the screen (only valid for grayscale devices)
screen.x, screen.y, screen.width, screen.height	Integer	The position and size of the screen, relative to the upper left corner of the device
screenPaintableRegion.x, screenPaintableRegion.y, screenPaintableRegion.width, screenPaintableRegion.height	Integer	The position and size of the paintable region (exclusive soft buttons), relative to the upper left corner of the screen.
canvasPaintableRegion.x, canvasPaintableRegion.y, canvasPaintableRegion.width, canvasPaintableRegion.height	Integer	The position and size of the paintable region of a Canvas object (exclusive soft buttons), relative to the upper left corner of the screen. Only values differing from the screenPaintableRegion are required
softbutton.<number>	x, y, w, h, font, align	Soft button rendering position, font, and alignment. Please note: Soft button counting starts with 0 here (instead of 1 for all other properties)

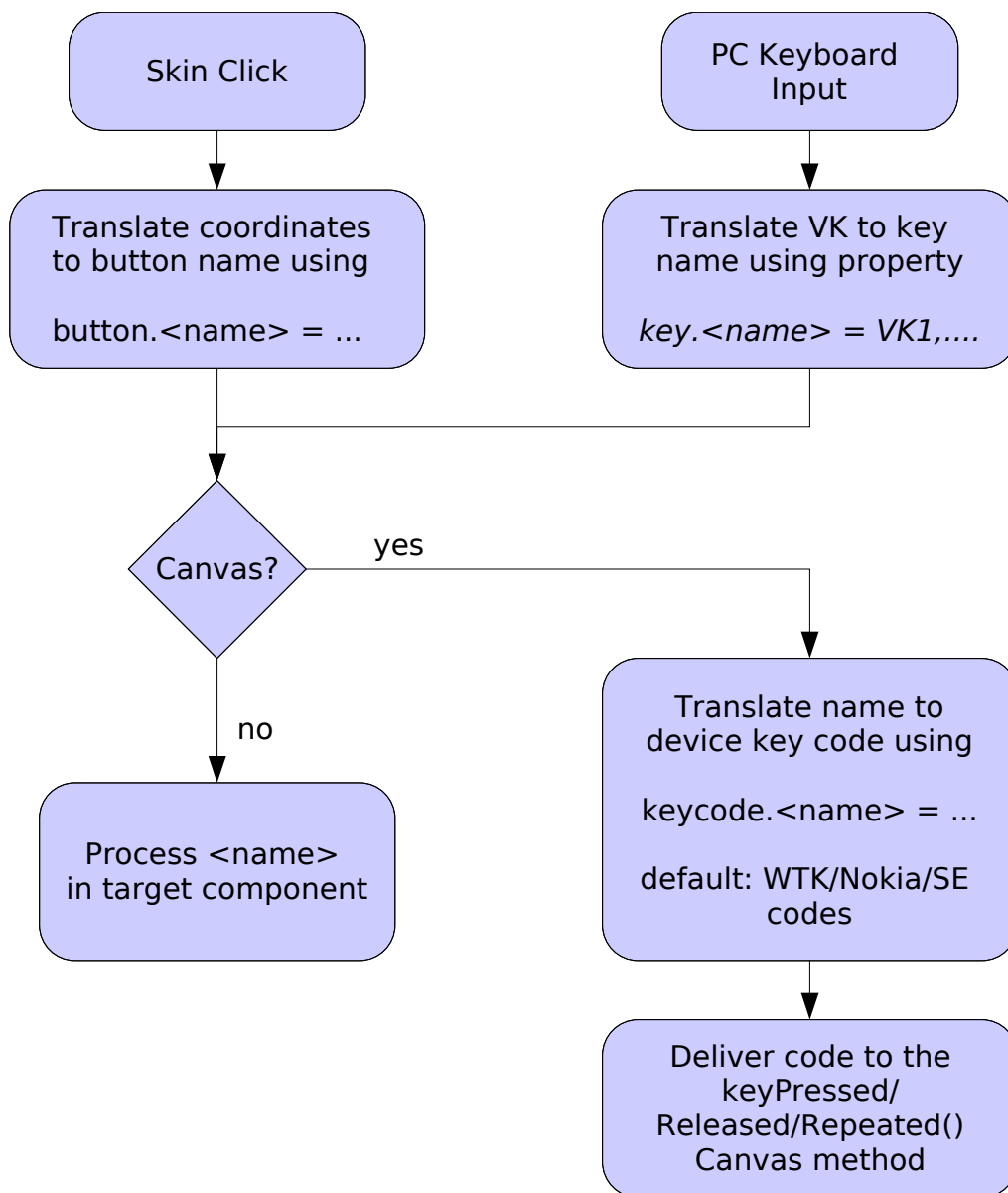
## 5 Device Key Processing

The processing of regular device keys is controlled by the skin properties *button.<name>*, *key.<name>*, and *keycode.<name>*, as illustrated in Figure 1.

Property	Value/Type	Description
button.<button_name>	x, y, width, height	Rectangular area in the skin image that is associated with the button <button_name>
keycode.<button_name>	<device keycode>	Assigns a (decimal) device key code to a button
key.<button_name>	<virtual key codes>	Assigns PC Keyboard key codes to a button.
game.<name>	<button_name>	Assigns a game action to a device button. Valid game action names are: LEFT, RIGHT, UP, DOWN, SELECT (SELCT corresponds to Canvas.FIRE)

Several key names are associated with special semantics and a default keyboard mapping:

Key Name	Default Keyboard Mapping	Default Device Key Code	Description
CLEAR	VK_BACKSPACE, VK_DELETE	-8	Special “clear” or “C” key available on some devices
SELECT	VK_ENTER	-5	
SOFT1 ... SOFT8	VK_F1 ... VK_F8	-6, -7	Soft keys
BACK	VK_ESCAPE		“Back” key available on SonyEricsson devices
UP	VK_UP		cursor up
DOWN	VK_DOWN		cursor down
LEFT	VK_LEFT		cursor left
RIGHT	VK_RIGHT		cursor right



getGameAction() performs an inverse lookup via keycode.<name>, then translates the name to a game action via the property

gameaction.<name> = <name>

Figure 1: Key Processing

## 6 Softkey and Command Handling and Localization

Please note that the on screen display of soft keys (softkey.<N>) is described in Section 2.1 “Display Properties”.

Property	Value/Type	Description
command.keys.<type>	<soft button name>	Set the preferred soft button for the command type <type> (ITEM, OK, ...).
command.menu.activate	<button name>	Allows to define the button that shows the command list.
command.menu.select	<button name>	Allows to define the button that selects a menu item. If the select command is a soft button, a "select" soft button will be available in lists.
options.menu.title	string	The title of the command menu. Default value is "Menu"
options.command.label	string	The label of the command to activate the command menu
cancel.command.label	string	Label for cancel commands (used to cancel the command menu)
ok.command.label	string	Label for ok commands (currently not used)

## 7 Internet Connections

In the default context, an Applet is not allowed to connect to other servers than the server hosting the Applet. Thus, for MIDlets using network connections, it may be necessary to set up a proxy server. The address of the proxy server is set in the skin definition file.

Property	Value/Type	Description
http.proxy	<host>:<port>	Address of the HTTP Proxy in the format "hostname:port"

## 8 System Properties

Property	Value/Type	Description
microedition.locale	String	iso 639 country code
microedition.encoding	String	default character encoding scheme
microedition.encoding.supported	String	supported character encoding schemes
http.proxy	<host>:<port>	Address of the HTTP Proxy in the format "hostname:port"

## 9 Font Configuration

Property	Value/Type	Description
font.<face>.<style>.<size>	<font definition>	defines a font that can be used in a MIDP canvas. The font definition may be a the name of a font properties file or a font name, style and size seperated in the form <face>.<style>.<size>, e.g. Arial-italic-9
font.default	<font definition>	Denotes the default font.
font.softButton	<font definition>	Denotes the soft button font.
font.<item type>	<font definition>	Denotes the font use for the given item type. Please refer to the section "Item Customization" for available item types.

## 10 Item Customization

Property	Value/Type	Description
<type>.background	RGB integer (e.g. 0xffffffff)	defines the foreground color of the given type of items, where type is one of <ul style="list-style-type: none"><li>• choiceGroup</li><li>• dateField</li><li>• default</li><li>• gauge</li><li>• item</li><li>• label</li><li>• textField</li><li>• textBox</li><li>• ticker</li></ul> which can be also be selected. To define the focus font add .<focus> Items which cannot be focused are <ul style="list-style-type: none"><li>• softButton</li><li>• title</li></ul>
<type>.foreground	RGB integer (e.g. 0xffffffff)	defines the foreground color of the given type of items
<type>.height	integer	Item height in pixel
<type>.align	left / right / center	Item alignment.
<type>.decoration	String	Item decoration (decoration draws a border), The following types are possible <ul style="list-style-type: none"><li>• shadow</li><li>• light</li><li>• underline</li><li>• overline</li><li>• decoration</li><li>• round</li><li>• border</li><li>• underlinechars</li><li>• compact</li></ul>
font.<type>	font definiton	Specifies the font for the given Item type (see font section)

## 11 Debugging and MIDlet tweaking

Property	Value/Type	Description
me4se.eventlog	true/ <b>false</b>	log calls to serviceRepaints(), paint(), keyPressed and other interactions between the emulator and the MIDlet
me4se.pause	<b>true</b> /false	Pauses the MIDlet when the applet is stopped. Can be disabled if the MIDlet has problems with subsequent calls to startApp()
me4se.restart	<b>true</b> /false	Allow to restart the MIDlet when it was terminated. Disable this Option for MIDlets that cannot be restarted.