Porting BlackBerry Storm applications and services to S60 5th Edition

Introduction

This article explains how to port applications from the BlackBerry Storm device to a S60 5th Edition device (Touch UI enabled device). This initial version compares the BlackBerry Storm and the Nokia 5800 XpressMusic device.

Considering that the native language of BlackBerry devices is Java, this article will focus initially at porting a Java application from the BlackBerry Storm to a Java application on S60 5th Edition. Then we will talk about porting the same application Symbian C++.

BlackBerry Storm - features

- Tactile feedback support
- Multi-touch support
- Capacitive touch-screen
- Screen size is 480 x 360
- 65,000 colors
- Accelerometer
- Input methods are stylus and finger touch support for text input and UI control (Multi-Tap, SureType in portrait orientation, QWERTY in landscape orientation)
- CPU clock rate is 528 MHz

Nokia 5800 XPressMusic - features

- Tactile feedback support
- Resistive touch-screen
- Screen size is 640 x 360
- 16 million colors
- Accelerometer
- Input methods are stylus, plectrum and finger touch support for text input and UI control (alphanumeric keypad, full and mini QWERTY keyboard, handwriting recognition)
- CPU clock rate is 369 MHz

(C) Copyright Nokia 2013. All rights reserved.

Capturing screen events in BlackBerry Storm - RIM APIs

High-level RIM APIs

The vast majority of high-level components in BlackBerry Storm will be touch-enabled, so you do not need to worry about the details.

The traditional input methods (Trackwheel, Trackball, Keypad) are handled using listeners or callback methods. Below we have some examples:

- Listeners
  - net.rim.device.api.ui.FocusChangeListener
  - net.rim.device.api.system.KeyListener
  - net.rim.device.api.ui.FieldChangeListener

- Callbacks
- keyChar(char character, int status, int time)
- trackwheelClick(int status, int time)
- navigationClick(int status, int time)

Regarding the touch input handling on BlackBerry Storm, the right way to handle the events is to use the **protected boolean net.rim.device.api.ui.Field.touchEvent(TouchEvent message)** callback event and not any listener. The sections below will give more detail regarding this method and the **net.rim.device.api.ui.TouchEvent** event.

Low-level RIM APIs

On Storm you can capture screen events by overriding the method **protected boolean touchEvent(TouchEvent message)** of class net.rim.device.api.ui.Field.

It is important to remember that the **net.rim.device.api.ui.Screen** class has the Field class as its ancestor, so one might have overriden this method on a subclass of Screen or any other class that extends one of its subclasses (like FullScreen).

There are also some BlackBerry specific MIDP classes that the application you are porting might have used. They are BlackBerryGameCanvas, BlackBerryCanvas, and BlackBerryCustomItem.

The **net.rim.device.api.ui.TouchEvent** is the same for both RIM native APIs as well as RIM MIDP APIs, so the original developer may have used the same methods for both approaches.

The TouchEvent class has methods that help you determine the coordinates where the touch ocurred, like the ones below:

- public abstract int getX(int touch)
- public abstract int getY(int touch)
- public abstract int getGlobalX(int touch)
- public abstract int getGlobalY(int touch)
- getMovePointsSize()
- getMovePoints(int, int[], int[], int[])
- getTime()
- getGesture()

To determine the type of event they may have used:

- public abstract int getEvent()

The valid events and their description is below:

- DOWN - Finger touch the screen
- UP - Finger(s) lifted from screen
- MOVE Finger drag or slide on screen
- CANCEL - Overriding system event, interrupting touch sequence
- GESTURE - Gesture detected, there is a method called getGesture() cite above
- UNCLICK - Finger releases from depressed screen until feedback is felt
- CLICK - Finger presses screen until feedback is felt

The net.rim.device.api.ui.TouchGesture class has a getEvent() and the values returned are:

- CLICK_REPEAT - Finger click and hold (keeping the pressure)
- TAP - Finger touch and then lift
- HOVER - Finger touch and hold
- SWIPE - Quick single-finger drag and lift move

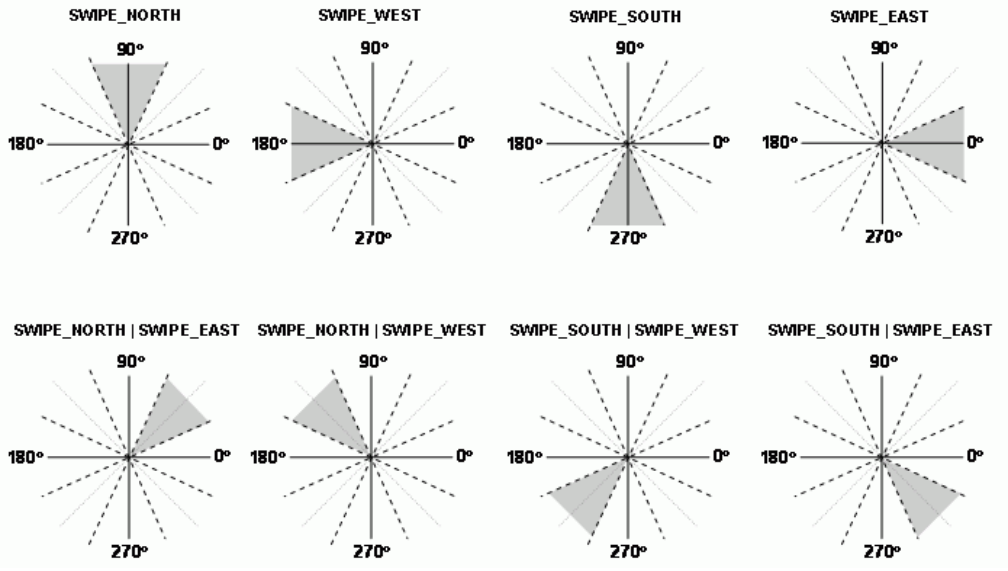
This class also has the methods below used to help with the return value of getEvent(). If it is **SWIPE** then you can:

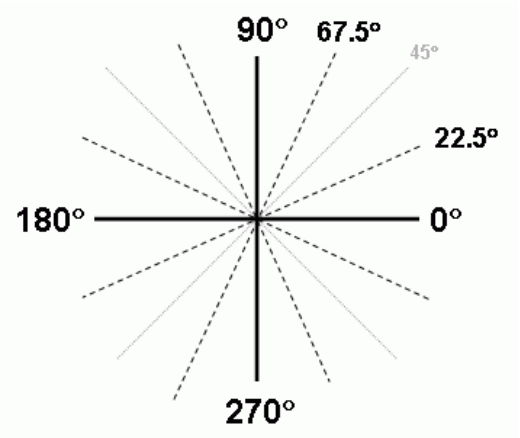
- getSwipeAngle() - Retrieves the angle (in degrees) associated with a swipe gesture relative to the device's current upward direction. For instance, 90 degrees always refers to the side of the display closest to the top of the device.
- getSwipeMagnitude() - Retrieves the magnitude (in pixels) associated with a swipe gesture.
- getSwipeDirection() - Retrieves the relative cardinal direction associated with a swipe gesture based on the device's upward direction. That is, SWIPE_NORTH always refers to the side of the display closest to the top of the device. Swipe gestures that occur at 90 degree intervals (North, South,

 (C) Copyright Nokia 2013. All rights reserved.

West, East) are +/- ~23 degrees. For example, SWIPE_NORTH is returned for (67, 112], where 67 is exclusive, and 112 is inclusive. For swipes occurring at a 45 degree interval +/- ~23 degrees, the two directions are bitwise ORed, so 135 degrees returns SWIPE_NORTH | SWIPE_WEST.

Some picture below help with the understanding and usage of the methods above.





If it is **HOVER** then you can:

- getHoverCount() - Retrieves the number of HOVER events generated before the user moves or removes touch from the touchscreen. A new consecutive HOVER event is generated every 100 milliseconds.

If it is **CLICK_REPEAT** then you can get:

- getClickRepeatCount() - Retrieves the number of CLICK_REPEAT events generated before the user moves or releases from the touchscreen. A new consecutive CLICK_REPEAT event is generated every 500 milliseconds.

If it is **TAP** then:

- getTapCount() - Retrieves the consecutive number of TAP events generated before the user moves or maintains touch for greater than 150 milliseconds.

Details you must take into consideration are:

- The TouchEvent object exists for the duration of event loop and expires;
- - There is a method called isValid() of net.rim.device.api.ui.TouchEvent class
  - So the right way to use it is to cache the required values, never the object

On the other hand, TouchGesture objects are immutable.

Virtual Keyboard

You use the net.rim.device.api.ui.Screen class and its getVirtualKeyboard() method to get it. Its class is net.rim.device.api.ui.VirtualKeyboard

You can control the visibility using the methods below:

- setVisbility(int)
- getVisibility()

The valid int values are:

- HIDE
- HIDE_FORCE
- IGNORE
- RESTORE

(C) Copyright Nokia 2013. All rights reserved.

- SHOW
- SHOW_FORCE

Screen orientation and direction

The class net.rim.device.api.ui.Display has a method called getOrientation().

The valid orientation values are:

- ORIENTATION_SQUARE
- ORIENTATION_LANDSCAPE
- ORIENTATION_PORTRAIT

The class net.rim.device.api.ui.UiEngineInstance has a method called getAcceptableDirections(int flags).

The valid direction values are (may be combined):

- net.rim.device.api.system.Display.DIRECTION_NORTH
- net.rim.device.api.system.Display.DIRECTION_SOUTH
- net.rim.device.api.system.Display.DIRECTION_EAST
- net.rim.device.api.system.Display.DIRECTION_WEST
- net.rim.device.api.system.Display.DIRECTION_LANDSCAPE
- net.rim.device.api.system.Display.DIRECTION_PORTRAIT

To restrict the directions valid for an application you can use the setAcceptableDirections(int) method.

To change the direction based on rotation you can use the protected abstract void sublayout(int width, int height) method of net.rim.device.api.ui.Manager class.

Capturing screen events on S60 - MIDP standard APIs (LCDUI and Canvas)

High-level MIDP APIs (LCDUI)

The vast majority of high-level components in MIDP will be touch-enabled on S60, so you do not need to worry about the details. LCDUI components will then automatically support touch events by default. Just link their events to the right event listener class. The standard navigation keys (UP, DOWN, LEFT, RIGHT and FIRE) will then be enabled as well. To parameter value to perform it explicitly must be set as below:

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
Nokia-MIDlet-On-Screen-Keypad: navigationkeys
```

MIDP 2.0 Game API

All the GameCanvas static fields (game actions) will be supported and available using the on-screen game oriented keypad (navigation keys plus FIRE, GAME_A, GAME_B GAME_C and GAME_D) that can be made available explicitly by specifying a JAD parameter as below:

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql,

 (C) Copyright Nokia 2013. All rights reserved.

povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
Nokia-MIDlet-On-Screen-Keypad: gameactions
```

In order to remove the on-screen keypad, provide a value of no as below:

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief, oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql, povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg, robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic, tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml, xorg_conf, xpp, yaml, z80, zxbasic

```
Nokia-MIDlet-On-Screen-Keypad: no
```

Low-level MIDP APIs (Canvas and M3G)

When porting, on your **javax.microedition.lcdui.Canvas** subclass you will override the pointerXXX() methods below:

- protected void pointerPressed(int x, int y)
- protected void pointerReleased(int x, int y)
- protected void pointerDragged(int x, int y)

As you can see, on standard MIDP Canvas the x and y coordinates are passed to the pointerXXX() methods, so you do not need to query them.

At last, to check exactly which events are supported, you can use the methods below:

- public boolean hasPointerEvents()
- public boolean hasPointerMotionEvents()

If using the **javax.microedition.lcdui.CustomItem** approach, besides the methods above you can use the **protected final int getInteractionModes()** to get the POINTER_PRESS, POINTER_RELEASE, and POINTER_DRAG values. Insert non-formatted text here

When using the Mobile 3D API (M3G) the same events above can be used to process and change the scene graph. The **javax.microedition.m3g.Group** class has a method called **public boolean pick(int scope, float x, float y, Camera camera, RayIntersection ri)** that picks the first Mesh or scaled Sprite3D in this Group that is enabled for picking, is intercepted by the given pick ray, and is in the specified scope.

The pick ray is cast from the given point p = (x, y) on the near clipping plane towards the corresponding point on the far clipping plane, and then beyond, from a pointer press event.

Touch support in M2G and SVG

When using the Scalable 2D Vector Graphics API for J2ME (M2G) the same pointer events above are generated, at first.

But the pointer events will not automatically delivered to M2G, so the developer must forward the events to M2G using the **public void dispatchMouseEvent(java.lang.String type,int x, int y)** of SVGImage class.

Invalid language.

You need to specify a language like this: `<source lang="html4strict">...</source>`

Supported languages for syntax highlighting:

4cs, 6502acme, 6502kickass, 6502tasm, 68000devpac, abap, actionscript, actionscript3, ada, algol68, apache, applescript, apt_sources, asm, asp, autoconf, autohotkey, autoit, avisynth, awk, bascomavr, bash, basic4gl, bf, bibtex, blitzbasic, bnf, boo, c, c_loadrunner, c_mac, caddcl, cadlisp, cfdg, cfm, chaiscript, cil, clojure, cmake, cobol, coffeescript, cpp, cpp-qt, csharp, css, cuesheet, d, dcs, delphi, diff, div, dos, dot, e, ecmascript, eiffel, email, epc, erlang, euphoria, f1, falcon, fo, fortran, freebasic, fsharp, gambas, gdb, genero, genie, gettext, glsl, gml, gnuplot, go, groovy, gwbasic, haskell, hicest, hq9plus, html4strict, html5, icon, idl, ini, inno, intercal, io, j, java, java5, javascript, jquery, kixtart, klonec, klonecpp, latex, lb, lisp, llvm, locobasic, logtalk, lolcode, lotusformulas, lotusscript, lscript, lsl2, lua, m68k, magiksf, make, mapbasic, matlab, mirc, mmix, modula2, modula3, mpasm, mxml, mysql, newlisp, nsis, oberon2, objc, objeck, ocaml, ocaml-brief,

(C) Copyright Nokia 2013. All rights reserved.

```
oobas, oracle11, oracle8, oxygene, oz, pascal, pcre, per, perl, perl6, pf, php, php-brief, pic16, pike, pixelbender, pli, plsql, postgresql,
povray, powerbuilder, powershell, proftpd, progress, prolog, properties, providex, purebasic, pycon, python, q, qbasic, rails, rebol, reg,
robots, rpmspec, rsplus, ruby, sas, scala, scheme, scilab, sdlbasic, smalltalk, smarty, sql, systemverilog, tcl, teraterm, text, thinbasic,
tsql, typoscript, unicon, uscript, vala, vb, vbnet, verilog, vhdl, vim, visualfoxpro, visualprolog, whitespace, whois, winbatch, xbasic, xml,
xorg_conf, xpp, yaml, z80, zxbasic
```

```
public void pointerPressed(int x, int y) {
img.dispatchMouseEvent("press", x, y);
}
```

The tne DOM Level 2 events such as DOMActivate are generated afterwards.

At last, the application can be notified which **org.w3c.dom.svg.SVGElement** has been clicked by means of its class that implements the **org.w3c.dom.events.EventListener** and the **void handleEvent(Event evt)** method.

Touch support in eSWT

All eSWT components should support touch events by default.

The **org.eclipse.ercp.swt.mobile.Screen** class has a method called **public boolean isTouchScreen()** method that allows you to check if touch screen events are supported.

Besides, you can call the **public void addEventListener(ScreenListener listener)** method to register a listener that implements the **org.eclipse.ercp.swt.mobile.ScreenListener** interface and overrides its screenXXX(ScreenEvent event)methods to be notified of events such as activation, deactivation and orientation changes. The ScreenEvent helps you further by providing access to orientation information, status and other widget information.

The **org.eclipse.swt.events** package has several listeners that can be implemented to detect and handle events.

As a touch ui example, if you implment the **org.eclipse.swt.events.MouseListener** interface and override its **void mouseDoubleClick(MouseEvent e)**method, the MouseEvent will give you some information regarding the touch event like the y and x coordinates and the widget that originated the event.

Touch support in Nokia UI API 1.2 onwards

Starting from S60 5th edtion devices, they may support touch events, so tactile feedback extensions are provided by the Nokia UI API 1.2 onwards. Tactile feedback consists of simple indicators such as audio feedback or vibration that happen along touch events.

This is supported by the **com.nokia.mid.ui.TactileFeedback** class and work with both LCDUI and eSWT components.

It can be used in custom components and is automatic by default in high-level components.

Java APIs documentation (Javadocs - packages)

S60 5th Edition - Java ME

BlackBerry JDE 4.7.0 API Reference

(C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.