

XTD - 1: Tutorial

2001. 11. 23

XTD Document Version History

Version	Date	Author	Description
1.0	2001 - 11 - 23	Kim.K.C	Created.
1.1	2001 - 12 - 17	Kim.W.G	Modified. (HelloWorld)
1.1	2002 - 01 - 28	Kim.W.G	Modified. (Example Link)
1.2	2002 - 02 - 25	Kim.W.G	Modified. (Example / Image)

Table of Contents

MIDP.....	3
SK - VM	3
MIDP	3
SK - VM	4
J2ME API	5
SK - VM	6
LCDUI.....	7
.....	7
SK - VM	8
SK - VM : HelloWorld.....	8
SK - VM : HelloWorld + Command.....	10
SK - VM : Animation.....	12
SK - VM : Stock.....	14
SKT Service API.....	21
.....	22
.....	23
.....	26
.....	27
.....	28
SMS	29
.....	31
Device	32
FileIO.....	32
3D	34
Reference.....	37
About XTD	37
XCE Technical Document (XTD)	37

MIDP

가 . SK

SK-VM

가 .

VM(가) , SK Telecom WAP GVM SK-

. Mobile

가 , SK Telecom

SK-VM

SK-VM Java 2 Micro Edition(J2ME) Clean Room

. SK-VM CLDC

M-Configuration(가) MIDP M-Profile(가), SK

Telecom OEM-Specific Class(SKT-Profile) 가 API .

, SK-VM M-Configuraion

M-Profile, SKT-Profile 가 . SK-VM

M-Configuraion, M-Profile, SKT-Profile,

SK-VM

. SK-VM SK

Telecom clean room J2ME(Java 2 Micro Eidtion)

가 UI, , IO

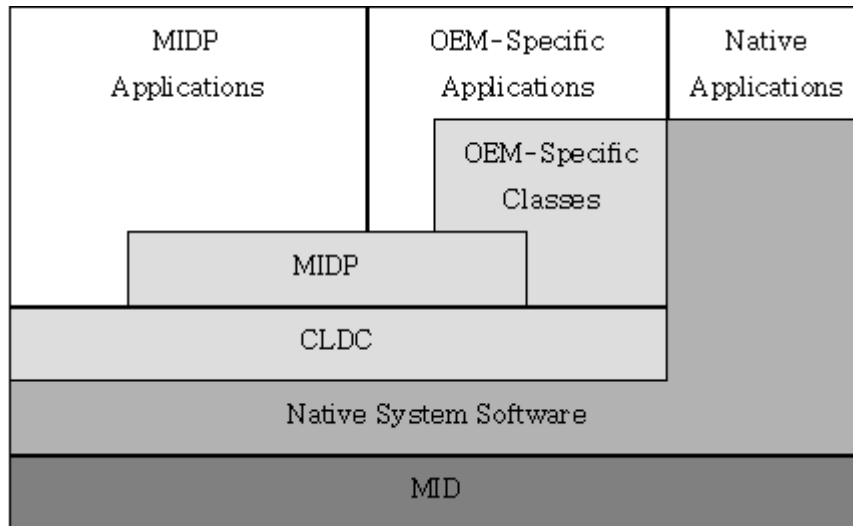
,

. MIDP SK-VM

J2ME API .

MIDP

가 CLDC(Connected, Limited Device Configuration) MIDP(Mobile Information Device Profile) Java 2 Micro Edition (J2ME) . J2ME .



[1] J2ME

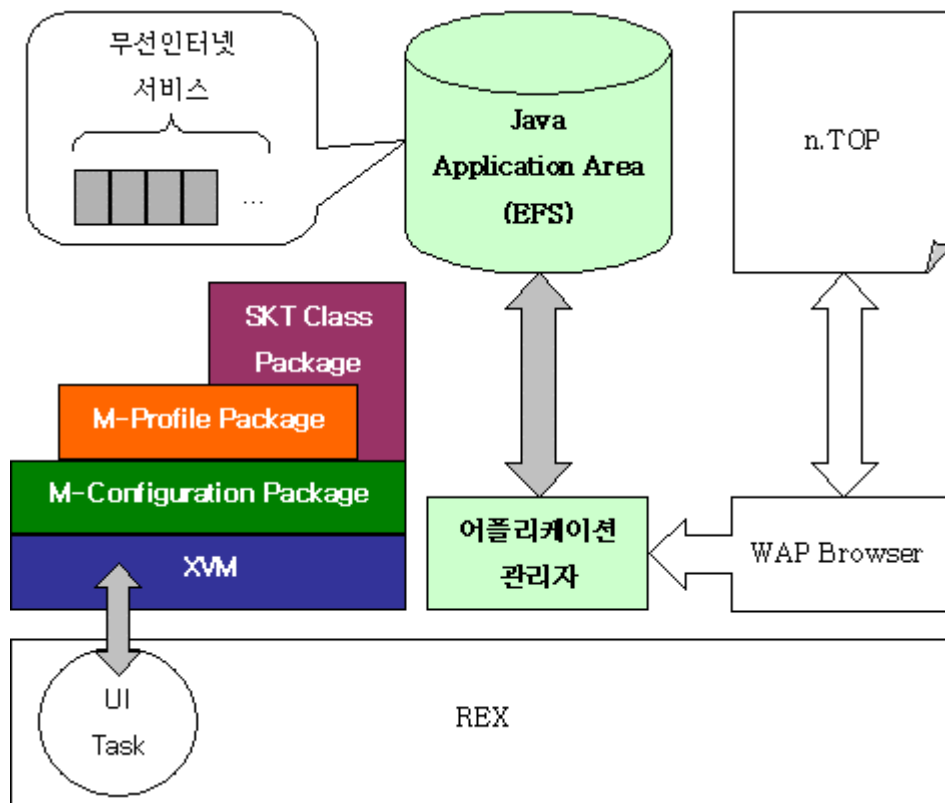
MID Mobile Information Device , Native System Software OS . Native System Software CLDC MIDP, OEM-Specific Classes API가 , API 가 .

- MIDP Applications : CLDC MIDP
- OEM-Specific Applications : CLDC MIDP 가 (OEM-specific Classes OEM-specific service API)
- Native Applications : J2ME WAP .

SK-VM

SK Telecom SK-VM

가 .



[2] SK Telecom SK-VM

가 , J2ME SK-VM KVM Clean Room Implementation , M-Configuration Package CLDC , M-Profile Package MIDP . SKT Class Package SK Telecom API J2ME OEM-Specific Classes .

J2ME API

J2ME CLDC MIDP API 가 . CLDC MIDP 가 .

Core Package

- Java.io :
- Java.lang : J2SE java.lang
- Java.util : J2SE java.util

Application Lifecycle Package

Javax.microedition.midlet : MIDP

User Interface Package

Javax.microedition.lcdui : MIDP UI

Persistence Package

Javax.microedition.rms : , ,

Networking Package

JavaX.microedition.io :

SK - VM

```

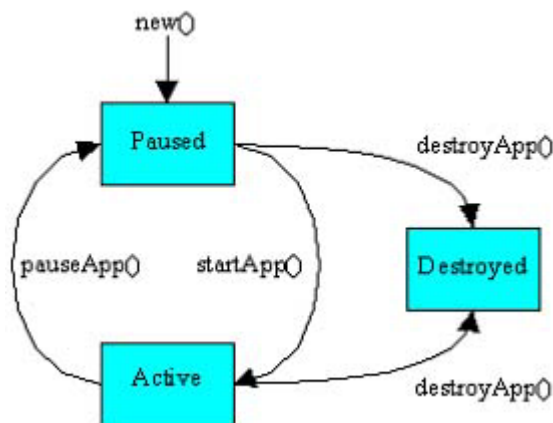
MIDP
, UI
,
.
javax.microedition.midlet
,
javax.microedition.lcdui
,
javax.microedition.io
. MIDP
, Mobile
.

```

```

MIDP . MIDP MIDlet
      . MIDlet Paused, Active, Destroyed
      . [ 3] .

```



[3] MIDlet Lifecycle Model

Paused :

Active :

Destroyed :

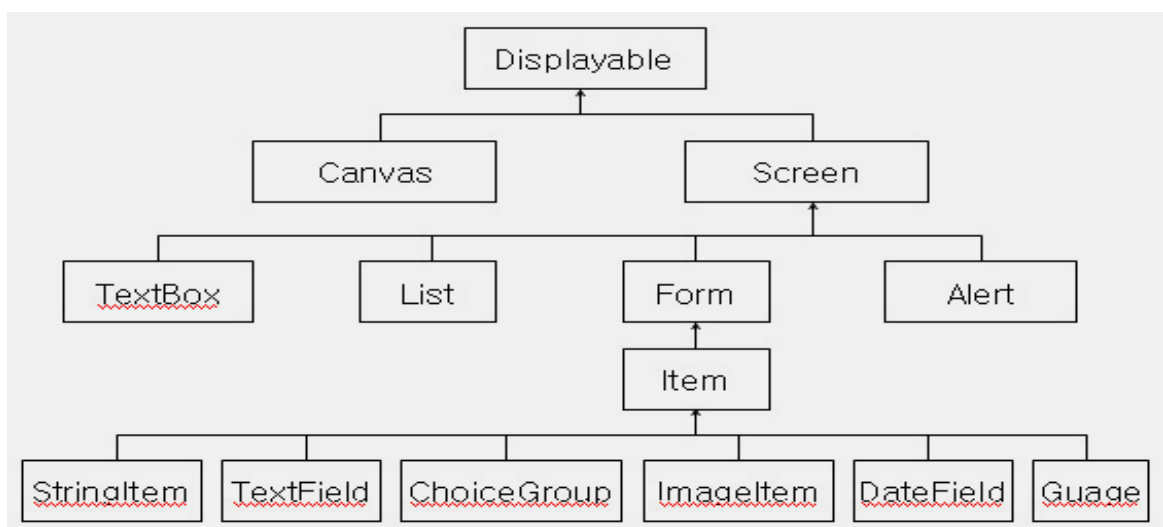
LCDUI

MIDP LCDUI
[4] . MIDP UI

: MIDP Screen

Display.setCurrent(Displayable)

2 : MIDP 2 API
. TextBox, Form
API . API UI
. Canvas, Graphics API

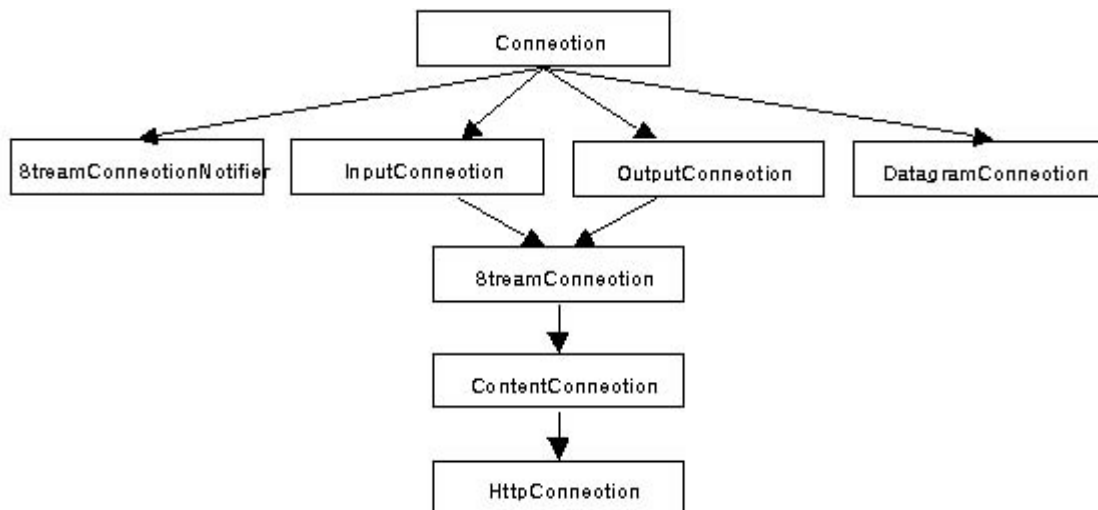


[4] LCDUI

MIDP [5]

. Connection Connector


```
static Connection open(String name)
name      : {schema}:{[target]}[{parms}]
Scheme    , target    , parms
";x=y"    .
```



[5] Connection Interface

SK - VM

MIDP 가 .
 SK - VM , MIDP . MIDP
 , Mobile 가
 HelloWorld ,
 ,
 .
 mobile
 가 ,
 가 .

SK - VM : HelloWorld

가 HelloWorld .

```

class HelloWorld {
public:
    HelloWorld() {}
    ~HelloWorld() {}
    void startApp(), pauseApp(),
    destroyApp()
};

class TextBox {
public:
    TextBox() {}
    ~TextBox() {}
    void startApp(), pauseApp(),
    destroyApp()
};

class MIDP {
public:
    MIDP() {}
    ~MIDP() {}
    void startApp(), pauseApp(),
    destroyApp()
};

```

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

/* MIDP          MIDlet          . */

public class HelloWorld extends MIDlet
{
    private Display display;
    private TextBox tb;

    public HelloWorld()
    {
        /*          Display          . */
        display = Display.getDisplay(this);

        /*          :
        TextBox(String title, String text, int maxSize, int constraints)
        "Example" : TextBox
        "Hello World" : TextBox
        20 : TextBox
        TextField.ANY : TextBox
        */
        tb = new TextBox("Example", "Hello World", 20, TextField.ANY);
    }

    public void startApp()
    {
        /*          가 Displayable 가          . */
        display.setCurrent(tb);
    }

    public void pauseApp() {}

    public void destroyApp(boolean bool) {}
}
```

}

< >
HelloWorld


[6] HelloWorld

SK-VM : HelloWorld + Command

HelloWorld
"exit"
Command , CommandListener

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

/* MIDP          MIDlet          . */
public class HelloWorld extends MIDlet implements CommandListener
{
    private Display display;
    private TextBox tb;
    private Command exit;

    public HelloWorld()
    {
        /*          Display          . */
        display = Display.getDisplay(this);
    }
}
```

```

        exit = new Command("EXIT", Command.EXIT, 0);

        /*
         *
         * TextBox(String title, String text, int maxSize, int constraints);
         *
         * "Example" : TextBox
         * "Hello World" : TextBox
         * 20 : TextBox
         * TextField.ANY : TextBox
         */

        tb = new TextBox("Example", "Hello World", 20, TextField.ANY);
        tb.addCommand(exit);
        tb.setCommandListener(this);
    }

    public void startApp()
    {
        /*
         * 가 Displayable 가 . */
        display.setCurrent(tb);
    }

    public void pauseApp(){}
    public void destroyApp(boolean bool){}
    public void commandAction(Command c, Displayable d)
    {
        if(c == exit){ destroyApp(false);notifyDestroyed(); }
    }
}

```

< >
 "exit"가 가 HelloWorld .



[7] HelloWorld + Command

SK-VM : Animation

```

MIDP
    . 5
    Animation
    .
    Animation startApp() AniCanvas . AniCanvas
    Runnable run() callSerially()
    repaint()
    ,
    .

```

```

import java.io.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Animation extends MIDlet{
    private Display display;
    public Animation(){
        display = Display.getDisplay(this);
    }

    public void startApp(){
        /* AniCanvas , . */
        display.setCurrent(new AniCanvas(display));
    }

    public void pauseApp(){}
    public void destroyApp(boolean bool){}
}

class AniCanvas extends Canvas implements Runnable{
    private static final int IMG_NUM = 5;
    private Display display;
    private Image image[];
    private int image_order;

```

```

public AniCanvas(Display display){
    this.display = display;
    image = new Image[IMG_NUM];

    try{
        /*          5          */
        image[0] = Image.createImage("/image1.lbm");
        image[1] = Image.createImage("/image2.lbm");
        image[2] = Image.createImage("/image3.lbm");
        image[3] = Image.createImage("/image4.lbm");
        image[4] = Image.createImage("/image5.lbm");
    }catch (IOException ioe) {}
    image_order = 0;

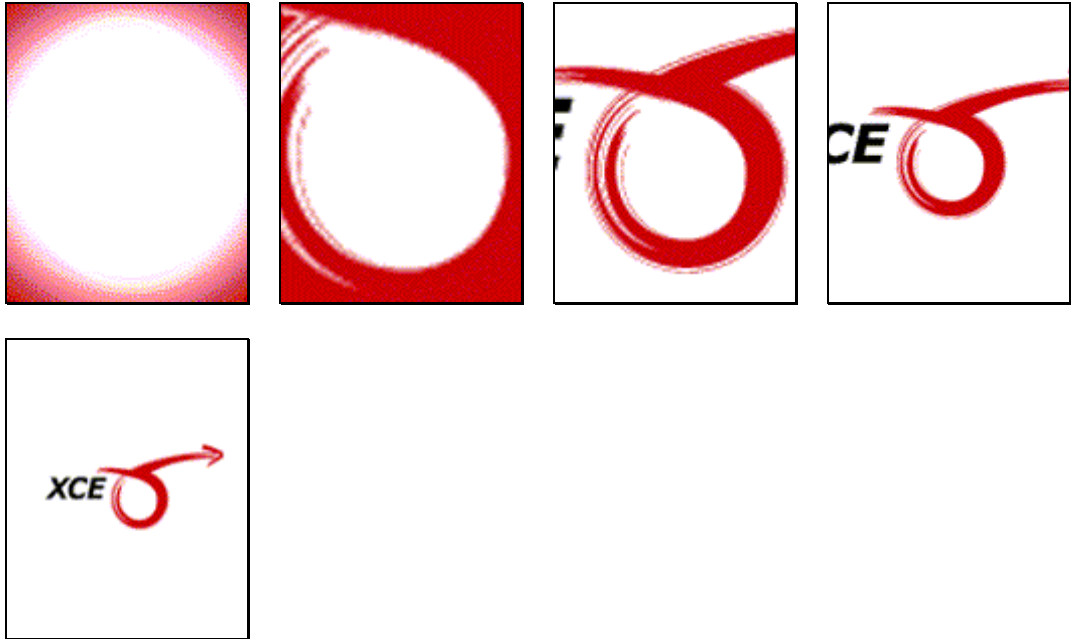
    /*          */
    Thread t = new Thread(this);
    t.start();
}

public void run(){
    /* repaint()   가   run()   */
    display.callSerially(this);
    repaint();
}

public void paint(Graphics g){
    /*          */
    g.drawImage(image[image_order],0,0,Graphics.TOP|Graphics.LEFT);
    image_order = (image_order+1)%IMG_NUM;
}
}

```

< >
가 .



[8] Animation

SK-VM : Stock

HTTP

4가

Key

Key

가

- Conf : Color

- StockMIDlet : MenuCanvas

- MenuCanvas : 가 가

InterestCanvas

- InterestCanvas : HTTP 가

가

```
<Conf >
package Stock;
public class Conf{
```

```

        public static final int BLACK = 0;

        public static final int DK_GRAY = 64;

        public static final int LT_GRAY = 128;

        public static final int WHITE = 255;

        public Conf(){}
    }

<StockMIDlet          >
package Stock;

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import java.io.*;

public class StockMIDlet extends MIDlet{

    private Display display;
    private MenuCanvas mc;

    public StockMIDlet(){
        display = Display.getDisplay(this);

        /* MenuCanvas          . */
        try { mc = new MenuCanvas(display); }
        catch (IOException ioe) {}
    }

    public void startApp(){
        display.setCurrent(mc);
        mc.repaint();
    }

    public void pauseApp() {}
    public void destroyApp(boolean bool){}
}

<MenuCanvas          >

```



```

package Stock;

import javax.microedition.lcdui.*;
import java.io.*;

public class MenuCanvas extends Canvas{

    private Image off_bg, on_bg;
    private int selected_menu;
    private Display display;
    private InterestCanvas interest_cvs;

    public MenuCanvas(Display display) throws IOException{
        off_bg = Image.createImage("/rs/stock_img/on_bg.lbm");
        on_bg = Image.createImage("/rs/stock_img/off_bg.lbm");
        selected_menu = 0;

        this.display = display;
        interest_cvs = new InterestCanvas();
    }

    public void paint(Graphics g){
        g.setGrayScale(Config.WHITE);
        g.fillRect(0, 0, 128, 96);
        g.setGrayScale(Config.BLACK);
        switch(selected_menu){

        case 0:
            g.drawImage(on_bg, getWidth()/2, 20, Graphics.HCENTER|Graphics.TOP);
            g.drawImage(off_bg, getWidth()/2, 40, Graphics.HCENTER|Graphics.TOP);
            g.drawImage(off_bg, getWidth()/2, 60, Graphics.HCENTER|Graphics.TOP);
            g.drawString("      ", getWidth()/2, 22, Graphics.HCENTER|Graphics.TOP);
            g.drawString("      ", getWidth()/2, 42, Graphics.HCENTER|Graphics.TOP);
            g.drawString("      ", getWidth()/2, 62, Graphics.HCENTER|Graphics.TOP);
            break;

        case 1:

```

```

g.drawImage(off_bg, getWidth()/2, 20, Graphics.HCENTER|Graphics.TOP);
g.drawImage(on_bg, getWidth()/2, 40, Graphics.HCENTER|Graphics.TOP);
g.drawImage(off_bg, getWidth()/2, 60, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 22, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 42, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 62, Graphics.HCENTER|Graphics.TOP);
break;

case 2:
g.drawImage(off_bg, getWidth()/2, 20, Graphics.HCENTER|Graphics.TOP);
g.drawImage(off_bg, getWidth()/2, 40, Graphics.HCENTER|Graphics.TOP);
g.drawImage(on_bg, getWidth()/2, 60, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 22, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 42, Graphics.HCENTER|Graphics.TOP);
g.drawString("      ", getWidth()/2, 62, Graphics.HCENTER|Graphics.TOP);
break;
    }
}

public void keyPressed(int key){
    if(key == Canvas.DOWN){
        selected_menu = (selected_menu+1)%3;
        repaint();
    }
    else if(key == Canvas.UP){
        selected_menu = (selected_menu+2)%3;
        repaint();
    }
    else if(key == Canvas.FIRE){
        switch(selected_menu){
            case 0:
                try { interest_cvs.getData(); }
                catch (IOException ioe) {}
                /* InterestCanvas . */
                display.setCurrent(interest_cvs);
                interest_cvs.repaint(0, 0, 128, 96);

```

```

        break;

        case 1: break;

        case 2: break;

    }

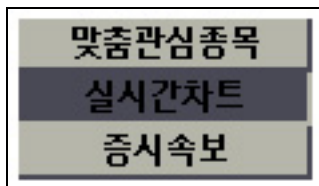
}

}

}

```

<MenuCanvas >



[9] MenuCanvas

```

<InterestCanvas >

package Stock;

import java.io.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;

public class InterestCanvas extends Canvas{

    /*      가      . */

    private final String INTEREST_STOCK[][] =

        {{ "17670", "SK      " }, { "05930", "      " }};

    private char price[][];

    public InterestCanvas(){

        price = new char[2][8];

    }

    public void paint(Graphics g){

        drawTable(g);

        drawData(g);

    }
}

```

```

public void getData() throws IOException{

    String url;

    StreamConnection sc = null;

    InputStream is = null;

    byte buf[] = new byte[128];

    for (int i = 0; i < 2; i ++){

        int len=0;

        try{

            /*
            HTTP
            URL stocktest.xce.co.kr
            */

            url = "http://stocktest.xce.co.kr/stocktest.cgi?s=" + INTEREST_STOCK[i][0];

            sc = (StreamConnection)Connector.open(url);

            is = sc.openInputStream();

            /*
            buf
            */

            is.read(buf);

        }

        finally{

            if (is != null) is.close();

            if (sc != null) sc.close();

        }

        boolean in_price = false;

        for (int j = 0, k = 0; j < buf.length; j++){

            if (in_price && ((char)(buf[j]) == '<')) break;

            if (in_price) price[i][k++] = (char)(buf[j]);

            if ((char)(buf[j]) == '>') in_price = true;

        }

    }

    /*
    */

    private void drawTable(Graphics g){

```

```

        g.setGrayScale(Config.WHITE);
        g.fillRect(0, 0, 128, 96);

        g.setGrayScale(Config.BLACK);
        g.drawLine(0, 20, 128, 20);
        g.drawLine(10, 30, 120, 30);
        g.drawLine(10, 50, 120, 50);
        g.drawLine(10, 70, 120, 70);
        g.drawLine(10, 90, 120, 90);

        g.drawLine(10, 30, 10, 90);
        g.drawLine(20, 30, 20, 90);
        g.drawLine(70, 30, 70, 90);
        g.drawLine(120, 30, 120, 90);

    }

    /*          가          가          . */
    private void drawData(Graphics g)
    {
        g.setGrayScale(Config.BLACK);

        g.drawString("          ", 64, 2, Graphics.HCENTER|Graphics.TOP);
        g.drawString("          ", 45, 32, Graphics.HCENTER|Graphics.TOP);
        g.drawString("   가", 95, 32, Graphics.HCENTER|Graphics.TOP);

        g.drawString("1", 15, 52, Graphics.HCENTER|Graphics.TOP);
        g.drawString(INTEREST_STOCK[0][1], 45, 52, Graphics.HCENTER|Graphics.TOP);
        g.drawChars(price[0], 0, 8, 95, 52, Graphics.HCENTER|Graphics.TOP);

        g.drawString("2", 15, 72, Graphics.HCENTER|Graphics.TOP);
        g.drawString(INTEREST_STOCK[1][1], 45, 72, Graphics.HCENTER|Graphics.TOP);
        g.drawChars(price[1], 0, 8, 95, 72, Graphics.HCENTER|Graphics.TOP);
    }
}

```

<InterestCanvas >

관심종목		
	종목명	현재가
1	SK텔레콤	266500
2	삼성전자	349000

[10] InterestCanvas

SK-VM

. , M-Profile UI
M-Configuration M-Profile
가 . Windows DirectX
가 . SK-VM
M-Profile ,
SKT API .

HelloWorld.java
Animation.java
Conf.java
StockMIDlet.java
MenuCanvas.java off_bg.gif on_bg.gif
InterestCanvas.java

SKT Service API

SKT Service API M-Configuration/M-Profile
API 가 . API
SKT API .
com.skt.m .
SKT Service API , , , , SMS ,

SKT API

SK-VM

MIDP

2D

. Graphics2D

```
void paint(Graphics g){
    ...
    Graphics2D g2 = Graphics2D.getGraphics2D(g);
    ...
}
```

Graphics2D

getGraphic2D()

. Graphics2D

getGraphics2D()

Graphics

Graphics

Graphics2D

getGraphics2D()

```
public void paint(Graphics g){
    g.drawImage(image1, 0, 40, Graphics.TOP | Graphics.LEFT);
    Graphics2D g2 = Graphics2D.getGraphics2D(g);
    g2.drawImage(10,60,image2,0,0,image2.getWidth(),image2.getHeight(),Graphics2D.DRAW
_COPY);
}
```

drawImage

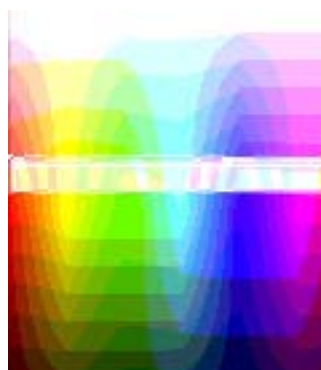
. drawImage

가
mode

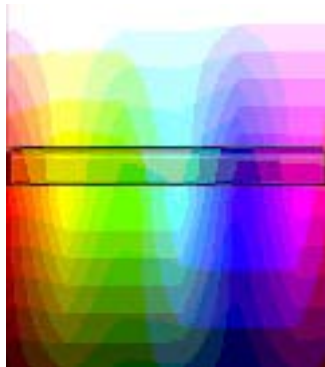
AND, OR, XOR



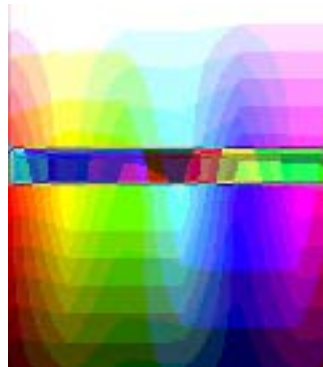
DRAW_COPY



DRAW_OR



DRAW_AND



DRAW_XOR

```
public void paint(Graphics g) {
    g.drawImage(image1, 0, 40, Graphics.TOP | Graphics.LEFT);
    Graphics2D g2 = Graphics2D.getGraphics2D(g);
    g2.invertRect(20,20,90,70);
}
```



captureLCD

,
getPixel, setPixel

createMaskableImage, getPixelMask, setPixelMask,

SK - VM MathFP
CLDC float double
double

MathFP long

MathFP

(MathTest.java)

```

System.out.println(
    "123.456 + 456.789 = "
    + MathFP.toStringLF(MathFP.add(
        MathFP.parseFPString("123.456"), MathFP.parseFPString("456.789")), 2
    )
);

```

```

123.456 + 456.789
    . parseFPString
toStringLF      double  10
                ,
                , toStringLF
                .
                ,
                double      long
parseFP( long ), double    String toStringE(long)
    .
    .

```

123.456

```
MathFP.round(MathFP.parseFPString("123.456"))
```

123.456

```
MathFP.parseFPString(MathFP.parseFPString("123.456"))
```

123.456 / 98.76

```
MathFP.divide(MathFP.parseFPString("123.456"), MathFP.parseFPString("98.76"))
```

123.456 * 98.76

```
MathFP.multiply(MathFP.parseFPString("123.456"), MathFP.parseFPString("98.76"))
```

123.456 - 98.76

```
MathFP.sub(MathFP.parseFPString("123.456"),MathFP.parseFPString("98.76"))
```

123.456

```
MathFP.abs(MathFP.parseFPString("123.456"))
```

123.456

```
MathFP.sqrt(MathFP.parseFPString("123.456"))
```

123.456 sine

```
MathFP.sin(MathFP.parseFPString("123.456"))
```

123.456 cosine

```
MathFP.cos(MathFP.parseFPString("123.456"))
```

123.456 cosine

```
MathFP.tan(MathFP.parseFPString("123.456"))
```

123.456, 98.76

```
MathFP.min(MathFP.parseFPString("123.456"),MathFP.parseFPString("98.76"))
```

123.456, 98.76

```
MathFP.max(MathFP.parseFPString("123.456"),MathFP.parseFPString("98.76"))
```

123.456 arc sine

```
MathFP.asin(MathFP.parseFPString("-98.76"))
```

-98.76 arc cosine

```
MathFP.acos(MathFP.parseFPString("-98.76"))
```

-98.76 arc tangent

```
MathFP.atan(MathFP.parseFPString("123.456"))
```

123.456

```
MathFP.exp(MathFP.parseFPString("123.456"))
```

123.456

```
MathFP.log(MathFP.parseFPString("123.456"))
```

123.456 4

```
MathFP.pow(MathFP.parseFPString("123.456"), MathFP.parseFPString("4"))
```

SK-VM

AudioSystem, AudioClip, AudioPlayer가

SK-VM AudioPlayer

AudioClip

(AudioClipTest.java)

```
try {
    AudioClip clip = AudioSystem.getAudioClip("mmf");
} catch (Exception e) { }
```

AudioClip . mmf AudioClip

mmf AudioClip AudioSystem getAudioClip()
AudioClip

UnsupportedFormatException

AudioClip

snd.mmf

```
try {
    AudioClip clip = AudioSystem.getAudioClip("mmf");
} catch (Exception e) { }
```

```
try {
    InputStream is = getClass().getResourceAsStream("snd.mmf");
    byte[] buffer = new byte[is.available()];
    clip.open(buffer, 0, buffer.length);
}
```

```
clip.play();
} catch (Exception e) {
} finally {
    try { clip.close(); } catch (Exception e) {}
}
```

AudioClip open() 가
 . AudioClip play()
 가 .

Note :

AudioClip play()
 block .

SK-VM

Vibration 가 가
 , 가 .

. ([VibTest.java](#))

```
public void keyPressed(int keyCode){
    switch (keyCode){
        case KEY_NUM0:
        case KEY_NUM1:
        case KEY_NUM2:
        case KEY_NUM3:
        case KEY_NUM4:
        case KEY_NUM5:
        case KEY_NUM6:
        case KEY_NUM7:
        case KEY_NUM8:
        case KEY_NUM9:
```

```

        Vibration.start(0, (keyCode - KEY_NUM0) * 1000);
        break;
    }
}

```

*1000 millisecond
 . start *static void start(int level, int timeout)*
 level level , timeout millisecond .

getLevelNum()

가 isSupported()

SK-VM

가
가

BackLight 가 .

. (BackLightTest.java)

```

public void keyPressed(int keyCode) {
    switch (keyCode) {
        case KEY_NUM0:
        case KEY_NUM1:
        case KEY_NUM2:
        case KEY_NUM3:
        case KEY_NUM4:
        case KEY_NUM5:
        case KEY_NUM6:
        case KEY_NUM7:
    }
}

```

```

        case KEY_NUM8:
        case KEY_NUM9:
            BackLight.on((keyCode - KEY_NUM0) * 1000);
            break;
        }
    }

```

```

        *1000 millisecond
        . on
        public static void on(int duration)
duration
        millisecond
        off()

        on, off
        getColor, setColor

SMS
SK - VM
SMS
SMS
, SMS
SMS
SMSListener
SMS
SMS Message
가
SMS
SMS

SK - VM
SMS Message
가
. (SMSTest.java )

SHORT_MESSAGE :
(SHORT_MESSAGE)

DOWNLOAD_NOTIFICATION :

APPLICATION_DATA :
Push

```

SHORT_MESSAGE

```
public class SMSTest extends MIDlet
implements SMSListener, CommandListener{
.....

    public void onMessage(SMSMessage msg){
        StringBuffer buffer = new StringBuffer();
        if(msg.getType() == SMSMessage.APPLICATION_DATA){
            alert.setTitle("APP_DATA    ");
            byte[] data= msg.getAppData();

            if (data != null){
                buffer.append("        : \n");
                buffer.append(new String(msg.getAppData()));
            }
        }
        alert.setString(buffer.toString());
        alert.setTimeout(Alert.FOREVER);
        display.setCurrent(alert);
    }
    .....
}
```

가

. SMS SMSListener

implements , onMessage

.

SMSMessage ApplicationData

public SMSMessage(String cname, byte data[]), 가 public

String getSender(), 가 public byte[] getAppData()가 .

SMS static public SMSMessage

get(int idx) , available()

.

[] onMessage

APPLICATION_DATA

```
SMSMessage msg = new SMSMessage(field3.getString().getBytes(), field2.getString());
SMS.send(field1.getString(), msg);
```

SMS message . SMS message
 SMS send(String receiver, SMSMessage msg)
 SMSMessage
 SMS Short Message , Download Notification,
 Application Data

SMSMessage Short Message *public SMSMessage(byte data[], String sender)* , Download Notification *public SMSMessage(String url, String name, String comment)* , Application Data
public SMSMessage(String cname, byte data[])

SK - VM .f
 Call
 (CallTest.java)

```
TextBox box = new TextBox("Call Test", "", 20,
    TextField.PHONENUMBER);
...
Call.connect(box.getString());
...
```

TextBox
 connect() 가
 가 .
 " 가 .
 call SK - VM 가 , call
 SK - VM

isSupported 가

Device

SK-VM

Device ,
WAP

```
if( Device.isBacklightEnabled() ){
    Device.setBacklightEnabled(false);
}
```

가 , application
isBacklightEnabled
setBacklightEnabled . true
, false

```
if( Device.isKeyToneEnabled() ){
    Device.setKeyToneEnabled(false);
}
```

가 ,
isKeyToneEnabled
setKeyToneEnabled . true
가 false

```
Device.invokeWapBrowser("http://www.xce.co.kr");
```

WAP

invokeWapBrowser

URL

FileIO

SK-VM

EFS

. EFS

(MSD, JAR) ,
 RecordStore .
 com.xce.io FileInputStream, FileOutputStream, XFile
 .
 (FileIOTest.java)

```
FileInputStream fis = new FileInputStream("testfile.txt");
int len = fis.available();
byte buf[] = new byte[len];
fis.read(buf, 0, len);
fis.close();
FileOutputStream fos = new FileOutputStream(wFile);
fos.write(buf, 0, len);
fos.close();
```

rFile wFile .
 FileInputStream , available
 FileOutputStream
 .
 FileInputStream , FileOutputStream
 , XFile 가 . XFile(String filename,
 int mode)constructor READ, WRITE READ_WRITE mode
 . available, read, write, close
 FileIOStream , flush, seek
 .
 XFile .

```
XFile fis = new XFile("testfile.txt", XFile.READ);
int len = fis.available();
byte buf[] = new byte[len];
fis.read(buf, 0, len);
fis.close();
```

```
XFile fos = new XFile("newfile.txt", XFile.WRITE);
fos.write(buf, 0, len);
fos.close();
```

, XFile JAR 가 . XFile(jarfilename, filename) constructor . write .

```
XFile fis = new XFile("resfile.jar", "testfile.txt");
int len = fis.available();
byte buf[] = new byte[len];
fis.read(buf, 0, len);
fis.close();
```

filesize, exists, unlink static .

```
if( XFile.exists(filename)){
    System.out.println("delete : " + filename + "("
        + XFile.filesize(filename) +
        "Byte)");
    XFile.unlink(filename);
}
```

XFile mkdir, rmdir, rmdir, readdir
, fsused, fsavail 가 .

```
XFile.mkdir(dirname); //directory
XFile.rmdir(dirname); //directory
XFile.rmdir(dirname); //directory file
XFile xfile = new XFile("test", XFile.READ_DIRECTORY);
String name;
while ( (name = xfile.readdir()) != null )
    System.out.println(name); //directory file
```

3D

SK-VM

3

API가

. Graphics3D

Object3D , Object3D 3
Class , Graphics3D Object3D

Graphics3D, Object3D 3D
. (Test3D.java)

```
int[] x = {40, 37, 30, 22, 20, 22, 29, 37, 34...}; //x
int[] y = {0, 7, 10, 7, 0, -7, -10, -7, 0, 7...}; //y
int[] z = {0, 0, 0, 0, 0, 0, 0, 0, -19, -18...}; //z
int[] i0 = {0, 8, 1, 9, 2, 10, 3, 11, 4, 12...};
//
int[] i1 = {8, 9, 9, 10, 10, 11, 11, 12, 12...};
//
```

3D object vertices Object3D instance .
Object3D(name) constructor setVertices, setTriangles
, addVertex, addTriangle vertex 가
.
x, y, z . , x[i], y[i], z[i]
. i0, i1, i2 3D object ,
x,y,z . , i0[i], i1[i], i2[i]
, i[i] j x[j],y[j],z[j] .
col[i] i .

SKT api 3 . LCD
LCD
. Object3D translate
.

```
public void showNotify(){
    Graphics3D.setZBufferEnabled(true);
}
public void hideNotify(){
    Graphics3D.destroyZBuffer();
}
```

```

Z-Buffer          setZBufferEnabled(true)      가
,                  destroyZBuffer      Z-Buffer가

```

```
public void run(){
    obj.rotate(az += aInc, el += eInc, 0);
    obj.scale(scale, scale, scale);
    repaint(0, 0, 128, 112);
}

public void keyPressed(int keyCode){
    int actionCode = getGameAction(keyCode);
    switch (keyCode){
        case Canvas.KEY_NUM3:
            scale <= 1;
            break;
        ...
    }
}
```

```

3D scale , 가
. 3 scale , 6
scale . Object3D scale
10 shift ( <<10 ) .
scale .

```

```
public void paint(Graphics g){
    ...
    Graphics3D.clearZBuffer();
    Graphics3D.render(g, obj);
    ...
}
```

```

Object3D가
render
Z - Buffer
clear
paint
.
, drawWireframe
Graphics3D
.
```

Reference

- J2ME Specification : <http://java.sun.com/j2me/>
- CLDC 1.0 : <http://java.sun.com/products/cldc>
- MIDP 1.0 : <http://java.sun.com/products/midp>
- WAP 2.1
- TIA/EIA/IS-95(A) : Mobile Station-Base Station Compatibility Standard for Dual-mode Wideband Spread Spectrum Cellular Systems
- TIA/EIA/IS-637 : Short Message Service for Dual-mode Wideband Spread Spectrum Cellular Systems
- TIA/EIA/IS-707 : Data Service Option for Wideband Spread Spectrum System
- TIA/EIA/IS-99 : Data Service Option Standard for Wideband Spread Spectrum Digital Cellular System
- TIA/EIA/TSB-74 : Support for 14.4kbps Data Rate and PCS Interaction for Wideband Spread Spectrum Cellular System

About XTD

XCE Technical Document (XTD)

XCE

XCE Technical Document (XTD)

PDF

XTD-1001

XTD-4

XTD-1: Tutorial

SK-VM

CLDC/MIDP/SKT Service API

. 가

XTD-2: SDK User's Guide

SDK

SDK

XTD-3: SKT Service API Reference

Javadoc [html](#) .

XTD-4: LBM Format Specification

SK-VM [LBM](#) .

XTD-5: SK-VM History

SK-VM .

XTD-6: Supported Devices

SK-VM . (WAP WML
, , , User Agent Field,
,)

XTD-7: Provisioning the Contents

WML, MSD, JAR .

XTD-8: FAQ

FAQ [PDF](#) .

XTD-9: Dictionary for XTD

XTD .

XTD-10: Creating new TextComponent

TextComponent [TextComponentHandler](#) .

XTD-11: User Agent Field

User Agent Field .

XTD-12: Preparing the Media Files

User Agent Field .

XTD-1001: SK-VM Specification

SK-VM .

XTD-1002: SK-VM Porting Guide

SK-VM .

XTD-1003: SK-VM Compatibility Test Suite

.