## i アプリライブラリ カスタムパネル

ドコモの i-mode 携帯電話で動作する i アプリをビジネスに変える! そんなビジュアル性の高いコンポーネントをライブラリという形で無償提供しています。

本ライブラリを用いると、Panel で使えるようなコンポーネント(ラベル、テキストボックス、リ ストなど)を、Canvas 上で作成する事ができるようになります。

【実行画面】		
×Tul i 📿	<b></b> p	
氏名	たろう	
職業	セールス 💌	
性別	◎男 ◎女	
お好きな商品をどうぞ		
□コンポ □のど飴 □教科書		
□マウス		
この情報は大切に保管されます。		
	登録	

カスタムパネルの概要(パンフレット): (PDF 形式)

## 【ダウンロード】

内容	Final Release
バイナリ (Jar)	16 Kbyte
APIリファレンス (JavaDoc)	115 Kbyte
ソース (Source)	30(Kbyte)

ライセンス

本製品はオープンソースであり、ライセンスは <u>CPL</u>です。本製品を別のiアプリライブラリ として使用するのではなく、別の製品(iアプリ)に使用・販売・インポートする場合におい て、(CPLのコントリビューションに該当しないため)本製品に変更または追加を加えたと しても、オープンソースにする必要もなく、オブジェクトコードを独自のライセンスで頒布で きます。

ライブラリの使い方

## 使い方は、とても簡単で、

1. 本ライブラリ (CPanel.jar) を以下の場所に置きます。

C:¥iDKDoJa5.1¥lib¥component

2. i アプリエミュレータを起動して、[ プロジェクト ]-[i アプリライブラリ ] から 本ライブラリに チェックを付けます。

以上で本ライブラリが利用できるようになります。

Sample

```
/*
* CostumePanel Sample.java
  */
import java.io.DataInputStream;
 import java.io.DataOutputStream;
import java.io.IOException;
 import iappli.lib.graphics.*;
import iappli.lib.io.*
import iappli.lib.ui.*
import com.nttdocomo.ui.lApplication;
import com.nttdocomo.ui.Canvas;
import com.nttdocomo.ui.Display;
import com.nttdocomo.ui.Graphics;
import com.nttdocomo.ui.Graphics;
import com.nttdocomo.ui.Medialmage;
import com.nttdocomo.ui.Medialmage;
/**
* サンプル実装です
  */
public class Sample extends IApplication {
    public void start() {
        Image im = null;
    }
}
              // 背景画像の取得
             // Fixed By Tana
try {
    // スクラッチパッドまたはサーバからダウンロードする場合
    // ・ ADF の PackageURL にダウンロード元サイトのURLを指定のこと
    // ("http://localhost/" は Apache 等 で自マシンからダウンロードする場合。)
    // ・ ADF の UseNetwork を http に設定のこと
    // ・ ADF の SPSize に画像の保存に必要な領域を指定のこと
    bute detail - SPControl mediaRead(0, 0, 0, this.getSourceURL()+"image.gif");
                    Medialmage mi = MediaManager.getImage(data);
                   // プロジェクト/res 配下から取得する場合 (注:jar サイズが大きくなる)
// ・プロジェクト/res 配下に画像ファイルを置くこと
Medialmage mi = MediaManager.getImage("resource:///image.gif");
11
                    mi.use();
             im = mi.getImage();
} catch (Exception e) {
                    // 画像が存在しない
                    im = null;
                    e.printStackTrace();
             Display.setCurrent(new SampleCPanel(im));
       }
}
 * カスタムパネルのメイン画面です
class SampleCPanel extends CostumePanel implements SPObject {
       String list1[] = {"男", "女"};
```

String list2[] = {"大工", "SE", "OL", "料理人", "セールス String list3[] = {"コンポ", "のど飴", "教科書", "マウス", "車 νス "}; "車 "}: CButton bt = new CButton("登録"); CTextBox  $tx = new CTextBox(" \sqcup \boxplus ", 10, 10);$ CCheckList ck = new CCheckList( CCheckList.RADIO\_BUTTON, list1. Graphics.getColorOfName(Graphics.BLACK), Graphics.getColorOfRGB(200, 200, 200, 200), Graphics.getColorOfName(Graphics.WHITE)); CPulldownList Is = new CPulldownList(list2, Graphics.getColorOfName(Graphics.BLACK) Graphics.getColorOfRGB(255, 255, 255, 200), Graphics.getColorOfName(Graphics.WHITE)); CCheckList cd = new CCheckList( CCheckList.CHECKBOX, list3. Graphics.getColorOfName(Graphics.BLACK) Graphics.getColorOfRGB(200, 200, 200, 200), Graphics.getColorOfName(Graphics.WHITE)); CBulletinBoard cb = new CBulletinBoard(null, "ご登録", Graphics.getColorOfName(Graphics.PURPLE) Graphics.getColorOfRGB(255, 255, 200, 200), Graphics.getColorOfName(Graphics.RED)); \* コンストラクタ public SampleCPanel(Image im) { this.im = im; try )/ スクラッチパッドからデータを復元 (前回押したボタン) SPControl.read(this, 0, 1); } catch (IOException e) { } } \* キャンバスに表示するために使用するメソッドです。<br> / public void paint(Graphics g) { CLabel Ib1 = new CLabel("氏名",0,0,-1); CLabel Ib2 = new CLabel("職業",0,0,-1); CLabel Ib3 = new CLabel("性別",0,0,-1); CLabel Ib4 = new CLabel("性別",0,0,-1); CLabel Ib4 = new CLabel("古好きな商品をどうぞ" Graphics.getColorOfName(Graphics.RED),0,-1); CLabel Ibx = new CLabel("この情報は大切に保管されます。", Graphics.getColorOfName(Graphics.BLACK), Graphics.getColorOfRGB(100, 100, 100, 100), Graphics.getColorOfName(Graphics.WHITE)); g.lock(); // 背景画像が存在する場合 if (im != null) { `g.drawlmagé(im, (Display.getWidth() - im.getWidth())/2, (Display.getHeight() - im.getWidth())/2); } // 背景画像を少しだけ透過する g.setColor(Graphics.getColorOfRGB(240, 240, 255, 230)); g.fillRect(0, 0, Display.getWidth(), Display.getHeight()); // テキスト lb1.show(g, 2, 1, true); tx.show (g, 12, 1, true); // プルダウンリスト lb2.show(g, 2, 3, true); ls.show (g, 12, 3, 15, true); // ラジオボタン Ib3.show(g, 2, 5, true); ck.show (g, 12, 5, 2, 4, true);

```
// チェックボックス
         Ib4.show(g, 2, 7, true);
cd.show(g, 2, 9, 3, 8, true);
         // ラベル
         lbx.show(g, 4, 13, true);
         // ボタン
         bt.show (g, 14, 16, true);
         // ブリテンボード
cb.show (g, 32, 0, 6, 1, true);
         // イベントの登録
addEvent(tx);
         addEvent(Is)
         addEvent(ck)
         addEvent(cd)
         addEvent(bt)
         g.unlock(true);
    }
     * イベントが通知された時に呼ばれる処理です。<br>
    /
public void processEvent2(CComponent c, int t, int p) {
    if (c == bt && t == CostumePanel.BUTTON_PRESSED) {
        // ボタンが押されたらダイアログ表示
        CDialog c1 = new CDialog();
        c1.setCanvas(this);
              try {
// スクラッチパッドへデータを保管
              SPControl.write(this, 0, 1);
} catch (IOException e) { }
              Display.setCurrent(c1);
         }
         / else if (c == tx && t == CostumePanel.TEXT_CHANGED) {
    else if (c == ls && t == CostumePanel.SELECTION_CHANGED) {
    else if (c == ck && t == CostumePanel.SELECTION_CHANGED) {
    else if (c == cd && t == CostumePanel.SELECTION_CHANGED) {
    }
}
    }
    /**
* スクラッチパッドから取得するデータ
    byte data[] = new byte[10];
         in.read(data);
         tx.setText(new String(data).trim());
         // 職業
         ls.select(in.readInt());
         return 14;
    }
    /**
* スクラッチパッドへ保存するデータ
    public int writeObject(DataOutputStream out) throws IOException {
         // 名前
byte data[] = new byte[10];
         String sdata = tx.getText();
System.arraycopy(sdata.getBytes(), 0, data, 0, sdata.getBytes().length);
         out.write(data);
                                                           // 10Byte
         // 職業
         out.writeInt(ls.getSelectedIndex()); // 4Byte
         return 14;
    }
    private Image im = null;
'* ダイアログ表示クラスです。<br>
* CBulletinBoard と CButton の組み合わせで作成しています。<br>

【実行例】
* CDialog c1 = new CDialog();
```

}

\* \*

```
* c1.setCanvas(this);
 * Display.setCurrent(c1);
* 
*/
class CDialog extends CostumePanel {
    CBulletinBoard cb = new CBulletinBoard("登録確認",
"¥n¥n この情報で登録してもよろしいですか?",
Graphics.getColorOfName(Graphics.BLACK),
Graphics.getColorOfRGB(255, 255, 200, 200),
Craphics.getColorOfRGB(255, 255, 200, 200),
            Graphics.getColorOfName(Graphics.WHITE));
    CButton bt1 = new CButton(" はい ");
CButton bt2 = new CButton("いいえ");
     * キャンバスに表示するために使用するメソッドです。<br>
     */
    public void paint(Graphics g) {
        g.lock();
        // 遷移元のキャンパスの画像を背景にする。
        g.drawlmage(im, 0, 0);
        // ブリテンボードの表示
        cb.show(g, 4, 3, 30, 11, true);
        // ボタンの表示
        bt1.show(g, 10, 12, false);
bt2.show(g, 22, 12, false);
addEvent(bt1);
        addEvent(bt2);
        g.unlock(true);
    }
    /**
     *
       遷移元のキャンパスを設定します。<br>
     *
       Display.setCurrent() する前に実行してください。
     *
       @param c 遷移元のキャンパスを指定します。
     * /
    public void setCanvas(Canvas c) {
        CS = C;
        // 遷移元のキャンパスの画像をバックバッファに取得する。
        }
     * イベントが通知された時に呼ばれる処理です。<br>
    public void processEvent2(CComponent c, int t, int p) {
        // ボタンのイベント
        if (t == CostumePanel.BUTTON_PRESSED) {
    // ボタン1が押された
            if (c == bt1) {
// 完了画面にやさしく遷移する
                Canvas c1 = new TestLabel();
                CCanvasFade.fade(CCanvasFade.FADE_BLENDIMAGE, this, c1, 5);
                im = null;
            }
// ボタン2が押された
            cb.clear();
im = null;
                Display.setCurrent(cs);
            }
        }
    }
    private Canvas cs = null;
private Image im = null;
                                 // 遷移元のキャンパス
// 遷移元のキャンパスイメージ
/**
* カスタムパネルの完了画面です。
*/
```

}

```
class TestLabel extends CostumePanel {
    /**
    * キャンバスに表示するために使用するメソッドです。<br>
    */
    public void paint(Graphics g) {
        CLabel lb = new CLabel("ご登録ありがとうございました。");
        g.lock();
        lb.show(g, 4, 8, true);
        g.unlock(true);
    }
}
```