**Swing 采用XML的方式自动布局的设想**

抗争的诗人 lnwazg@126.com

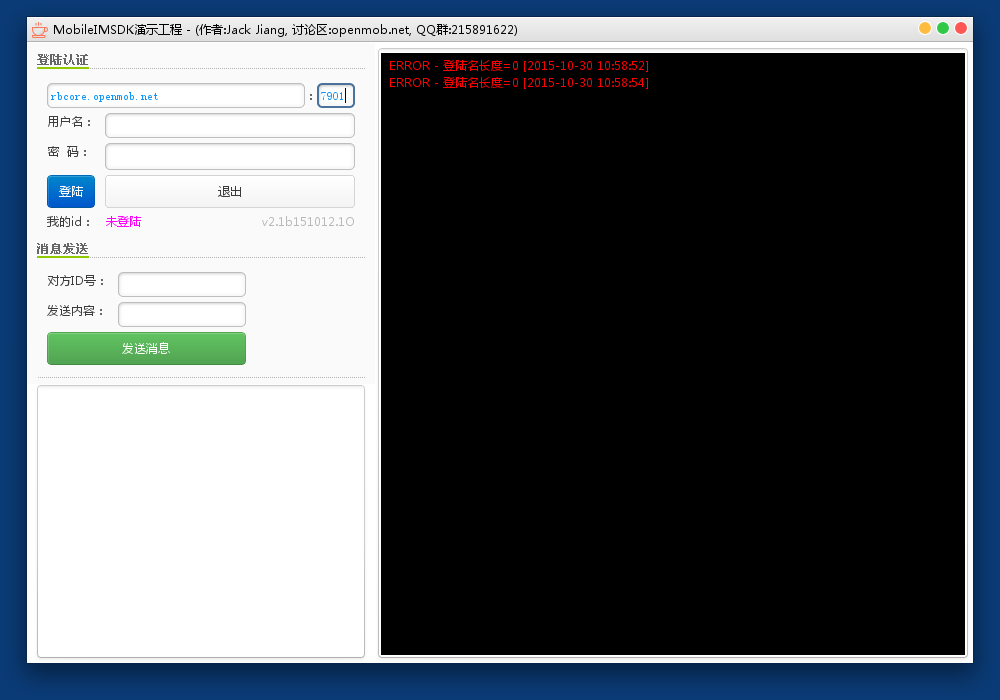
众所周知，java很强大，其强大性主要体现在成熟的社区组织、各类第三方开源的jar类库。所有的这些第三方的功能包，都极大地拓宽了java的适用范围。自动垃圾回收机制，让编写java应用变得轻松自然。

但桌面应用一直是java的软肋。长久以来，swing需要采用拖沓冗余的代码来实现复杂的界面布局、“重口味”的原生swing界面，均让Swing这一领域一直不愠不火。Swing界面刷新机制EDT里面的坑，均吓倒了一大批欲在这个领域做出点什么的新人。

个人认为，布局代码“不直观”，是摆在swing开发者尤其是面前的第一道大关。但是这个问题在安卓领域却早已拥有成熟的解决方案：采用xml方式布局。那么，比安卓历史更久的Swing技术为何还没有一个简单易用的xml布局工具？

举个例子：

为了实现以下的界面，



我们要写这么长的布局代码：

private void initViews()

{

this.editServerIp = new JTextField(16);

this.editServerPort = new JTextField(5);

this.editServerIp.setForeground(new Color(13, 148, 252));

this.editServerPort.setForeground(new Color(13, 148, 252));

this.editServerIp.setText("rbcore.openmob.net");

this.editServerPort.setText("7901");

this.btnLogin = new JButton("登陆");

this.btnLogin.setUI(new BEButtonUI().setNormalColor(BEButtonUI.NormalColor.blue));

this.btnLogin.setForeground(Color.white);

this.editLoginName = new JTextField(22);

this.editLoginPsw = new JPasswordField(22);

this.btnLogout = new JButton("退出");

this.viewMyid = new JLabel();

this.viewMyid.setForeground(new Color(255, 0, 255));

this.viewMyid.setText("未登陆");

this.btnSend = new JButton("发送消息");

this.btnSend.setUI(new BEButtonUI().setNormalColor(BEButtonUI.NormalColor.green));

this.btnSend.setForeground(Color.white);

this.editId = new JTextField(20);

this.editContent = new JTextField(20);

this.debugPane = new JTextPane();

this.debugPane.setBackground(Color.black);

this.debugPane.setCaretColor(Color.white);

Log.getInstance().setLogDest(this.debugPane);

this.imInfoPane = new JTextPane();

HardLayoutPane authPanel = new HardLayoutPane();

JPanel serverInfoPane = new JPanel(new BorderLayout());

JPanel portInfoPane = new JPanel(new BorderLayout());

portInfoPane.add(new JLabel("："), "West");

portInfoPane.add(this.editServerPort, "Center");

serverInfoPane.add(this.editServerIp, "Center");

serverInfoPane.add(portInfoPane, "East");

authPanel.addTo(serverInfoPane, 2, true);

authPanel.nextLine();

authPanel.addTo(new JLabel("用户名："), 1, true);

authPanel.addTo(this.editLoginName, 1, true);

authPanel.nextLine();

authPanel.addTo(new JLabel("密 码："), 1, true);

authPanel.addTo(this.editLoginPsw, 1, true);

authPanel.nextLine();

authPanel.addTo(this.btnLogin, 1, true);

authPanel.addTo(this.btnLogout, 1, true);

authPanel.nextLine();

authPanel.addTo(new JLabel("我的id："), 1, true);

JPanel idAndVerPanel = new JPanel();

idAndVerPanel.setLayout(new BoxLayout(idAndVerPanel, 2));

JLabel lbVer = new JLabel("v2.1b151012.1O");

lbVer.setForeground(new Color(184, 184, 184));

idAndVerPanel.add(this.viewMyid);

idAndVerPanel.add(Box.createHorizontalGlue());

idAndVerPanel.add(lbVer);

authPanel.addTo(idAndVerPanel, 1, true);

authPanel.nextLine();

HardLayoutPane toPanel = new HardLayoutPane();

toPanel.addTo(new JLabel("对方ID号："), 1, true);

toPanel.addTo(this.editId, 1, true);

toPanel.nextLine();

toPanel.addTo(new JLabel("发送内容："), 1, true);

toPanel.addTo(this.editContent, 1, true);

toPanel.nextLine();

toPanel.addTo(this.btnSend, 4, true);

toPanel.nextLine();

HardLayoutPane oprPanel = new HardLayoutPane();

oprPanel.addTitledLineSeparator("登陆认证");

oprPanel.addTo(authPanel, 1, true);

oprPanel.addTitledLineSeparator("消息发送");

oprPanel.addTo(toPanel, 1, true);

oprPanel.addTitledLineSeparator();

JPanel leftPanel = new JPanel(new BorderLayout());

leftPanel.add(oprPanel, "North");

JScrollPane imInfoSc = new JScrollPane(this.imInfoPane);

imInfoSc.setBorder(BorderFactory.createCompoundBorder(

BorderFactory.createEmptyBorder(0, 7, 0, 7), imInfoSc.getBorder()));

imInfoSc.setHorizontalScrollBarPolicy(31);

leftPanel.add(imInfoSc, "Center");

getContentPane().setLayout(new BorderLayout());

getContentPane().add(leftPanel, "West");

JScrollPane sc = new JScrollPane(this.debugPane);

sc.setBorder(BorderFactory.createCompoundBorder(

BorderFactory.createEmptyBorder(4, 0, 0, 2), sc.getBorder()));

sc.setHorizontalScrollBarPolicy(31);

getContentPane().add(sc, "Center");

setTitle("MobileIMSDK演示工程 - (作者:Jack Jiang, 讨论区:openmob.net, QQ群:215891622)");

setLocationRelativeTo(null);

setSize(1000, 700);

}

界面很漂亮，但是以上的代码长度让我不由得哼起来“这让我感到绝望，董小姐”。

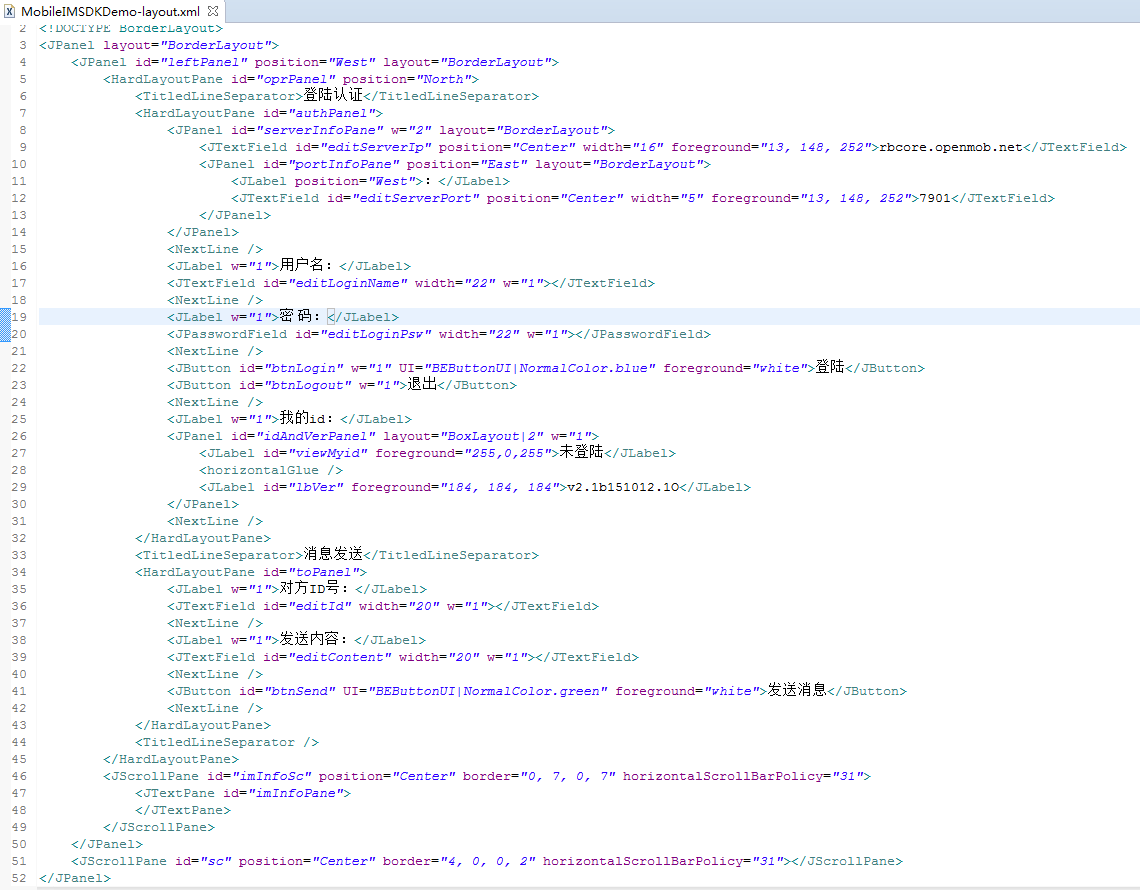
难道写swing界面的人，真的要变成码农么？

画界面这么低级、无意义、费时费力的事情，为何要我们一遍又一遍地靠写代码完成？

难道就没有一种更好的方式去解决这个问题吗？

安卓早已有更成熟的做法，为何不能直接拿来用？

于是，经过一夜的苦思冥想，我终于找到了下面的方式：



我将上述的复杂界面代码，转换成为一个完整的xml描述文件。 所有组件的属性，均可以在xml的节点中自由增加。配合自己写的高度定制的布局xml解析器，从此界面布局这一“Boring Zone”终于可以自动化啦！

终于，写Swing应用不再那么费时费力，而布局，将清晰呈现。配合采用反射技术写成的xml解析器，可以实现xml里面的组件ID与后台的组件声明自动绑定。写Swing程序，从此将和开发Web应用一样容易。今后的开发步骤基本为：

1. 写xml描述文件
2. 后台事件绑定

其基本原理，已经和web的jquery开发相当接近！加入组件双向绑定之后，其开发体验将和angularJS接近！

后台要获取一个组件，可以采用类似Jquery的方式：

JButton btnSend = (JButton)X$.get(“btnSend”);

绑定事件照旧这样写：

btnSend.addActionListener(new ActionListener(){

…

});

时间就是金钱，在用自动化改善我们的效率方面，我们JAVA程序员始终名列前茅！

附件：

