# How to configure an UI

1. **UI Construction**

A general UI includes three parts, top level frame, middle level page and inner component. That means a whole UI needs a window, panel and some non-container components.

For example, you can configure a whole UI as below

* Window

A window is the outer frame, it can be a Frame or Dialog. A xml file can represent a window, I designate its name as ‘xml\_window.xml’

<?xml version="1.0" encoding="UTF-8"?>

<window

xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'

xsi:noNamespaceSchemaLocation='WindowXmlSchema.xsd'> <!—window schema and namespace -->

<id>0</id> <!—window id, unique number , as top frame we should give its zero value -->

<type>Frame</type> <!—window type, value is Frame or Dialog -->

<ui-style> <!—window ui style setting part -->

<lookandfeel> <!—window lookandfeel name, if no this setting, default value will be used -->

com.jgoodies.looks.windows.WindowsLookAndFeel

</lookandfeel>

<font> <!—system default font setting, you can also set font in each component -->

<name>Fangsong</name>

<size>21</size>

<style>BOLD</style>

</font>

<background>/image/5.jpg</background> <!—system default background setting -->

<progress>/image/loader\_orange\_256.gif</progress> <!—system default progress animation setting -->

<color>115,164,209</color> <!—system default basic color setting -->

<textcolor>64,0,64</textcolor> <!—system default text color setting -->

</ui-style>

<driver> <!—system outer driver setting -->

<device>

<type>magnetCard</type> <!—device type -->

<port>com3</port> <!—device serial port -->

<dll>emptydll</dll> <!—device driver dll name -->

</device>

<device>

<type>ICCard</type>

<port>com3</port>

<dll>emptydll</dll>

</device>

<device>

<type>sensorCard</type>

<port>com3</port>

<dll>emptydll</dll>

</device>

<device>

<type>printer</type>

<port>com3</port>

<dll>emptydll</dll>

</device>

<device>

<type>keyboard</type>

<port>com3</port>

<dll>emptydll</dll>

</device>

</driver>

<attribute> <!—window attribute setting -->

<text>北京移动银行查询系统</text> <!—window title name -->

<icon>/image/bank-icon-256.png</icon> <!—window icon -->

</attribute>

<menubar> <!—window menu bar setting, window can have no menu depending on requirement -->

<menu>

<id>-10</id>

<name>管理操作</name>

<item>

<id>-11</id>

<name>用户管理</name>

<action>

<jumpAction>

<nextPanel>200</nextPanel>

</jumpAction>

</action>

</item>

<item>

<id>-12</id>

<name>设备管理</name>

<action>

<jumpAction>

<nextPanel>300</nextPanel>

</jumpAction>

</action>

</item>

</menu>

<menu>

<id>-20</id>

<name>关于</name>

<enable>true</enable>

<item>

<id>-21</id>

<name>版本信息</name>

<action>

<customizedAction>

<className>king.flow.action.customization.AboutAction</className>

</customizedAction>

</action>

</item>

</menu>

</menubar>

<contents> <!—window page setting-->

<page>./conf/100/panel-100.xml</page> <!—page link, if you wanna load panel in window, you should add a link of panel file here -->

<page>./conf/200/panel-200.xml</page>

<page>./conf/300/panel-300.xml</page>

<page>./conf/300/panel-330.xml</page>

<page>./conf/300/panel-360.xml</page>

<page>./conf/900/panel-900.xml</page>

<page>./conf/900/panel-970.xml</page>

</contents>

</window>

That’s a standard window



* Page

Page is really container component, which is loaded by window and contains all working components such as button, table , combobox and so on.

<?xml version="1.0" encoding="UTF-8"?>

<panel

xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'

xsi:noNamespaceSchemaLocation='../WindowXmlSchema.xsd'>

<id>100</id> <!—panel id, unique number -->

<type>Panel</type>

<active>true</active> <!—panel show status, true means this panel will show firstly, false means hide -->

<background>/image/5.jpg</background>

<component> <!— component setting -->

<id>101</id> <!— component unique id -->

<type>Label</type> <!— component type, current supported type are Button, Table, AdvancedTable, ComboBox, Label, TextField, PasswordField, TextArea, EditorPane, Date, NativeBroswer -->

<attribute> <!— component attribute -->

<rect> <!— component rectangle attribute -->

<x>600</x>

<y>0</y>

<width>600</width>

<heigh>150</heigh>

</rect>

<text>欢迎使用自助银行查询业务</text> <!— component text attribute -->

<icon>/image/5.ico</icon> <!— component icon attribute -->

<debug >false</debug> <!—debug attribute, true will show component border, false no border show -->

</attribute>

<action> <!— component action setting -->

<fontSetAction>

<fontSize>45</fontSize>

</fontSetAction>

</action>

</component>

<component>

<id>102</id>

<type>Button</type>

<attribute>

<text>查询业务</text>

<rect>

<x>200</x>

<y>300</y>

<width>300</width>

<heigh>50</heigh>

</rect>

<debug>false</debug>

</attribute>

<action>

<buttonJumpPanelAction>

<nextPanel>200</nextPanel>

</buttonJumpPanelAction>

</action>

</component>

<component>

<id>103</id>

<type>Button</type>

<attribute>

<text>启动/关闭虚拟键盘</text>

<rect>

<x>700</x>

<y>300</y>

<width>300</width>

<heigh>50</heigh>

</rect>

</attribute>

<action>

<fontSetAction>

<fontSize>18</fontSize>

</fontSetAction>

<virtualKeyboardAction>

<start>Display.exe</start>

<stop>Hide.exe</stop>

</virtualKeyboardAction>

</action>

</component>

<component>

<id>104</id>

<type>ComboBox</type>

<attribute>

<rect>

<x>700</x>

<y>400</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<comboShowAction>

<items>CNY/人民币, USD/美元, JPY/日元</items>

</comboShowAction>

<mediaPlayAction>

<media>./media/goodbye.wav</media>

</mediaPlayAction>

</action>

</component>

<component>

<id>105</id>

<type>PasswordField</type>

<attribute>

<rect>

<x>400</x>

<y>400</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<limitInputAction>

<length>6</length>

</limitInputAction>

<useTipAction>

<tip>&lt;html&gt;&lt;center&gt;&lt;h1&gt;Hello,&lt;/h1&gt;&lt;h2 style=&#39;color: red; font-style: italic;&#39;&gt;World&lt;/h2&gt;&lt;b&gt;Modified&lt;br/&gt;ToolTipText&lt;/b&gt;&lt;/center&gt;&lt;br/&gt;so simple.请输入账号密码</tip>

</useTipAction>

<mediaPlayAction>

<media>./media/song.wav</media>

</mediaPlayAction>

</action>

</component>

<component>

<id>106</id>

<type>TextField</type>

<attribute>

<rect>

<x>400</x>

<y>450</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<customizedAction>

<className>king.flow.action.customization.FingerPrintDriverAction</className>

<constructorParameters>

<parameter>

<type>boolean</type>

<value>true</value>

</parameter>

<parameter>

<type>int</type>

<value>104</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

</component>

<component>

<id>107</id>

<type>TextField</type>

<attribute>

<rect>

<x>400</x>

<y>500</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<customizedAction>

<className>king.flow.action.customization.CusorMoveAction</className>

<constructorParameters>

<parameter>

<type>int</type>

<value>106</value>

</parameter>

<parameter>

<type>int</type>

<value>105</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

</component>

</panel>

* Component

Current several types of component are supported here, including Button, Table, AdvancedTable, ComboBox, Label, TextField, PasswordField, Date

* Button

<component>

<id>102</id>

<type>Button</type>

<attribute>

<text>查询业务</text>

<rect>

<x>200</x>

<y>300</y>

<width>300</width>

<heigh>50</heigh>

</rect>

<debug>false</debug>

</attribute>

<action>

<buttonJumpPanelAction>

<nextPanel>200</nextPanel>

</buttonJumpPanelAction>

</action>

</component>



* ComboBox

<component>

<id>104</id>

<type>ComboBox</type>

<attribute>

<rect>

<x>700</x>

<y>400</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<comboShowAction>

<items>CNY/人民币, USD/美元, JPY/日元</items>

</comboShowAction>

<mediaPlayAction>

<media>./media/goodbye.wav</media>

</mediaPlayAction>

</action>

</component>



* Label

<component>

<id>101</id>

<type>Label</type>

<attribute>

<rect>

<x>600</x>

<y>0</y>

<width>600</width>

<heigh>150</heigh>

</rect>

<text>欢迎使用自助银行查询业务</text>

</attribute>

<action>

<fontSetAction>

<fontSize>45</fontSize>

</fontSetAction>

</action>

</component>



* TextField

<component>

<id>107</id>

<type>TextField</type>

<attribute>

<rect>

<x>400</x>

<y>500</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<customizedAction>

<className>king.flow.action.customization.CusorMoveAction</className>

<constructorParameters>

<parameter>

<type>int</type>

<value>106</value>

</parameter>

<parameter>

<type>int</type>

<value>105</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

</component>



* PasswordField

<component>

<id>105</id>

<type>PasswordField</type>

<attribute>

<rect>

<x>400</x>

<y>400</y>

<width>300</width>

<heigh>40</heigh>

</rect>

</attribute>

<action>

<limitInputAction>

<length>6</length>

</limitInputAction>

<useTipAction>

<tip>&lt;html&gt;&lt;center&gt;&lt;h1&gt;Hello,&lt;/h1&gt;&lt;h2 style=&#39;color: red; font-style: italic;&#39;&gt;World&lt;/h2&gt;&lt;b&gt;Modified&lt;br/&gt;ToolTipText&lt;/b&gt;&lt;/center&gt;&lt;br/&gt;so simple.请输入账号密码</tip>

</useTipAction>

<mediaPlayAction>

<media>./media/song.wav</media>

</mediaPlayAction>

</action>

</component>



* Date

<component>

<id>302</id>

<type>Date</type>

<attribute>

<rect>

<x>180</x>

<y>30</y>

<width>300</width>

<heigh>50</heigh>

</rect>

</attribute>

</component>



* Table

<component>

<id>311</id>

<type>Table</type>

<attribute>

<rect>

<x>0</x>

<y>100</y>

<width>300</width>

<heigh>450</heigh>

</rect>

<debug>false</debug>

</attribute>

<action>

<fontSetAction>

<fontSize>15</fontSize>

</fontSetAction>

</action>

<action>

<tableShowAction>

<columnNames>流水号, 名称, 开始时间, 结束时间, 费用</columnNames>

</tableShowAction>

</action>

</component>



If you need a scroll bar aside table, you need to put a table component in a decorator as below

**<decorator>**

**<id>310</id>**

**<type>ScrollPanel</type>**

**<attribute>**

**<rect>**

**<x>0</x>**

**<y>100</y>**

**<width>1600</width>**

**<heigh>600</heigh>**

**</rect>**

**<debug>false</debug>**

**</attribute>**

<component>

<id>311</id>

<type>Table</type>

<attribute>

<rect>

<x>0</x>

<y>100</y>

<width>300</width>

<heigh>450</heigh>

</rect>

<debug>false</debug>

</attribute>

<action>

<fontSetAction>

<fontSize>15</fontSize>

</fontSetAction>

</action>

<action>

<tableShowAction>

<columnNames>流水号, 名称, 开始时间, 结束时间, 费用</columnNames>

</tableShowAction>

</action>

</component>

**</decorator>**

* AdvancedTable

<component>

<id>330337</id>

<type>AdvancedTable</type>

<attribute>

<rect>

<x>0</x>

<y>95</y>

<width>1275</width>

<heigh>600</heigh>

</rect>

</attribute>

<action>

<tableShowAction>

<columnNames>流水号, 名称, 开始时间, 结束时间, 费用</columnNames>

</tableShowAction>

</action>

<action>

<msgSendAction>

<cmdCode>330</cmdCode>

<prsCode>queryAcount</prsCode>

<conditions>330332,330334</conditions>

<nextStep>

<nextPanel>330</nextPanel>

<display>330337</display>

</nextStep>

<exception>

<nextPanel>970</nextPanel>

<display>971</display>

</exception>

<checkRules>

<notEqual>

<more>330334</more>

<less>330332</less>

<errMsg>查询开始时间不能够大于结束时间</errMsg>

</notEqual>

</checkRules>

</msgSendAction>

</action>

</component>



1. **Action configuration**

After UI is constructed, you need to configure component action to deal with business requirements. There are several action type here,

* customizedAction

it used to load the third party plugin to do ad-hoc action.

* CusorMoveAction

It’s about cursor moving action, you can set the jumping previous and next components via up/down arrow button

<action>

<customizedAction>

<className>king.flow.action.customization.CusorMoveAction</className>

<constructorParameters> <!— constructor method parameters -->

<parameter> <!— the first constructor parameter, represents jumping previous component id -->

<type>int</type> <!—parameter type, int/string/Boolean/java.util.List/king.flow.view.ComponentEnum now supported -->

<value>106</value>

</parameter>

<parameter> <!— the second constructor parameter, represents jumping the next component id -->

<type>int</type>

<value>105</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

* FingerPrintDriverAction

It’s about finger printing action

<action>

<customizedAction>

<className>king.flow.action.customization.FingerPrintDriverAction</className>

<constructorParameters>

<parameter> <!— the first constructor parameter, represents whether it’s a finger print registration or reading a finger print saved. True is registry of finger print, false is reading print -->

<type>boolean</type>

<value>true</value>

</parameter>

<parameter> <!— the second constructor parameter, represents next jumping component after finger print operation is over -->

<type>int</type>

<value>104</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

* OpenBroswerAction

It’s about browser action

<action>

<customizedAction>

<className>king.flow.action.customization.OpenBroswerAction</className>

<constructorParameters>

<parameter>

<type>string</type>

<value>http://www.baidu.com</value>

</parameter>

</constructorParameters>

</customizedAction>

</action>

* DriverActiveAction

It’s about swipe card action

<action>

<customizedAction>

<className>king.flow.action.customization.DriverActiveAction</className>

<constructorParameters>

<parameter><!— the first constructor parameter, represents the next jumping component id after swiping card -->

<type>int</type>

<value>334</value>

</parameter>

<parameter><!— the second constructor parameter is optional, represents whether combobox is editable. Default value is true -->

<type>boolean</type>

<value>false</value>

</parameter>

</constructorParameters>

</customizedAction>

<comboShowAction>

<items>/, ACTION1/刷接触式卡, ACTION2/刷磁条卡, ACTION3/插入IC卡</items>

</comboShowAction>

<mediaPlayAction>

<media>./media/goodbye.wav</media>

</mediaPlayAction>

</action>

* jumpPanelAction

This action is about page jumping. that says if you click this button, I will switch the next page you wanna go.

<action>

<jumpPanelAction>

<nextPanel>200</nextPanel>

</jumpPanelAction>

</action>

* setFontAction

<action>

<setFontAction>

<fontSize>45</fontSize>

</setFontAction >

</action>

* cleanAction

when you leave some page, you hope to clean all data in that page, you can use this action.

<action>

<buttonJumpPanelAction>

<nextPanel>200</nextPanel>

</buttonJumpPanelAction>

<cleanAction>

<conditions>311,307, 337</conditions><!— when I jump to page200, please clean component 311,307 and 337 value -->

</cleanAction>

</action>

* limitInputAction

this action will limit input format in text field and password field component

<action>

<limitInputAction>

<length>6</length> <!— limit 6 characters of text component -->

</limitInputAction>

</action>

* useTipAction

give component tip show when cursor moves to this component

<action>

<limitInputAction>

<length>6</length>

</limitInputAction>

<useTipAction>

<tip>&lt;html&gt;&lt;center&gt;&lt;h1&gt;Hello,&lt;/h1&gt;&lt;h2 style=&#39;color: red; font-style: italic;&#39;&gt;World&lt;/h2&gt;&lt;b&gt;Modified&lt;br/&gt;ToolTipText&lt;/b&gt;&lt;/center&gt;&lt;br/&gt;so simple.请输入账号密码</tip> <!— tip content -->

</useTipAction>

<mediaPlayAction>

<media>./media/song.wav</media>

</mediaPlayAction>

</action>



* showTableAction

show a table in page

<action>

<showTableAction>

<columnNames>流水号, 名称, 开始时间, 结束时间, 费用</columnNames>

</showTableAction >

</action>



* showComboBoxAction

show a combobox in page

<showComboBoxAction>

<items>/, ACTION1/刷接触式卡, ACTION2/刷磁条卡, ACTION3/插入IC卡</items>

</showComboBoxAction >



* playMediaAction

Play a media when you move a cursor to this component

<action>

<playMediaAction>

<media>./media/goodbye.wav</media>

</playMediaAction>

</action>

* runCommandAction

run local command

<action>

<runCommandAction>

<command>notepad</command> <!— open notepad -->

</runCommandAction>

</action>



* virtualKeyboardAction

open Chinese character input tool

<virtualKeyboardAction>

<start>AVF\_Display.exe</start>

<stop>AVF\_Hide.exe</stop>

</virtualKeyboardAction>

* setPrinterAction

Setting receipt print output content

<action>

<setPrinterAction>

<header>通商银行</header>

<tail>此凭条仅供参照自助银行,如有异议请联系客服中心:96669</tail>

</setPrinterAction>

</action>

* sendMsgAction

Sending user input to server

<action>

<sendMsgAction>

<prsCode>queryAcount</prsCode> <!— operation prs code -->

<conditions>302,304,307,333,334,335</conditions> <!— data need to send up to server, and we need to catch those data from these components -->

<nextStep>

<nextPanel>300</nextPanel> <!— operation is successful, go to next page -->

<display>311</display> <!— in new page, show result in this component -->

</nextStep>

<exception>

<nextPanel>970</nextPanel> <!— operation is failed, go to error page -->

<display>971</display> <!— in error page, show result to user -->

</exception>

<checkRules> <!— rule checking before sending data to server -->

<notNull>

<content>333</content> <!— component input must not be empty -->

<errMsg>密码不能为空</errMsg>

</notNull>

<notNull>

<content>334</content>

<errMsg>确认密码不能为空</errMsg>

</notNull>

<notNull>

<content>335</content>

<errMsg>账户不能为空</errMsg>

</notNull>

<validateCJK>

<content>335</content> <!— component input must be Chinese character -->

<errMsg>账户名称必须是中文</errMsg>

</validateCJK>

<validateCJK>

<content>336</content>

<errMsg>输入必须是中文</errMsg>

</validateCJK>

<template>

<content>307</content> <!— component input must start with pattern value -->

<pattern>6222</pattern>

<errMsg>当前使用非本行发行卡片，请使用本行卡</errMsg>

</template>

<equal>

<conditions>333,334</conditions> <!— components input must be equal -->

<errMsg>两次输入的密码不一致,请重新输入</errMsg>

</equal>

<notEqual> <!— component input must not be equal -->

<more>304</more>

<less>302</less>

<errMsg>查询开始时间不能够大于结束时间</errMsg>

</notEqual>

</checkRules>

</sendMsgAction>

</action>

* insertICardAction

When you hit a button, activate IC card-inserting operation, you need to configure it

<action>

<insertICardAction>

<suceessfulPanel>370</suceessfulPanel> <!— insert a valid IC, jumping to next page -->

<failedPanel>200</failedPanel> <!— no valid IC found, jumping to a page-->

</insertICardAction>

</action>

* writeICardAction

just like sendMsgAction, you can configure this action to write data to IC card after a transaction

<action>

<writeICardAction>

<prsCode>buygas</prsCode>

<conditions>370004,370005,370006</conditions>

<checkRules>

<notNull>

<content>370004</content>

<errMsg>370004输入不能够为空</errMsg>

</notNull>

<notNull>

<content>370005</content>

<errMsg>370005输入不能够为空</errMsg>

</notNull>

<notNull>

<content>370006</content>

<errMsg>370006输入不能够为空</errMsg>

</notNull>

</checkRules>

<exception>

<display>9101</display>

<nextPanel>910</nextPanel>

</exception>

<nextStep>

<display>9101</display>

<nextPanel>910</nextPanel>

</nextStep>

</writeICardAction>

</action>