

**EUI详细设计说明**

V 1.1.01



**百富计算机技术(深圳)有限公司**

**PAX Computer Technology（Shenzhen）Co., Ltd.**

**富计算机**

版权声明

Copyright © 2000-2015 百富计算机技术（深圳）有限公司保留所有权利。未经百富计算机技术（深圳）有限公司书面许可,不得重新制作或散发本文档的任何内容。尽管百富计算机技术（深圳）有限公司努力确保本文档内容的准确性，但仍可能包含错误或遗漏。本文档所包含的信息如有更改，恕不另行通知。本文中的事例和样例程序仅用作说明演示目的，可能并不适用于您的要求，在您的软件投入商业使用前，请您仔细验证事例或样例程序的适用性。

修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本号** | **备注** | **作者** |
| 2015-10-14 | V1.0.0 | 编写初稿 | 陈晓键 |
| 2015-10-14 | V1.0.1 | 增加了两个UI库api接口，对UI的使用限制做出详细了说明，删减了单选框，复选框控件 | 陈晓键 |
| 2015-10-22 | V1.0.2 | 补全了控件的属性以及标出了控件属性的默认值，对一些页，控件等进行了解释,修改了部分语言表述问题，跟新了xml脚本应用示例,将图片修改为英文显示 | 丁雨成 |
| 2015-10-29 | V1.0.3 | PDKUI命名修改为EUI，包括接口名。修改部分语言的描述错误 | 陈晓键 |
| 2015-11-5 | V1.0.4 | 新增EUI\_Process函数返回值为3时的说明  删除是否斜体，是否粗体的属性描述  新增状态栏与非接灯资源文件名的详细说明 | 陈晓键 |
| 2015-11-11 | V1.0.5 | 修改菜单控件的行列间距名称分别为lineHeight,lineSpace。删除部分控件属性 | 陈晓键 |
| 2015-11-12 | V1.0.6 | 修改部分返回值定义 | 陈晓键 |
| 2015-11-16 | V1.0.7 | 详细说明各个控件属性的取值范围与默认值 | 陈晓键 |
| 2016-01-04 | V1.0.8 | 修改部分描述不正确的问题 | 陈晓键 |
| 2016-01-15 | V1.0.9 | 1. 修改部分描述 2. 新增虚拟全键盘功能 3. 新增多编辑框功能 | 陈晓键 |
| 2016-02-23 | V1.0.10 | 1. 修改部分描述 2. 修改EUI\_Init接口参数 3. 新增EUI\_Free接口 | 陈晓键 |
| 2016-03-11 | V1.0.11 | 1. 修改部分描述 | 陈晓键 |
| 2016-04-11 | V1.1.00 | 1. 修改编辑框textAlign属性的值为仅支持中左，居中，中右显示 2. EUI\_Process新增返回值，支持手势返回。 3. 页控件新增animationType属性，支持页面已某种动画效果呈现 4. 新增视频播放器控件 | 陈晓键 |
| 2016-04-22 | V1.1.01 | 1.新增复选框，单选框，网格，进度条控件的描述  2.删除视频播放器控件  3.新增返回值定义  4.删除page的animationType属性。  5.删除EUI\_Process接口返回值为8时的定义，即删除手势功能 | 陈晓键 |

目 录

[1 概述 7](#_Toc449080426)

[1.1 目的 7](#_Toc449080427)

[1.2 对象和范围 7](#_Toc449080428)

[1.3 适用平台 7](#_Toc449080429)

[1.4 书写约定 7](#_Toc449080430)

[1.5 术语 7](#_Toc449080431)

[2简述 9](#_Toc449080432)

[2.1 EUI设计思想 9](#_Toc449080433)

[3 EUI解决方案的组成 10](#_Toc449080434)

[3.1 UI工具 10](#_Toc449080435)

[3.2 xml文件 11](#_Toc449080436)

[3.3 UI库 12](#_Toc449080437)

[3.3.1 UI库组成 12](#_Toc449080438)

[3.3.2 UI的使用限制 12](#_Toc449080439)

[4 页与控件 14](#_Toc449080440)

[4.1页 14](#_Toc449080441)

[4.2控件 15](#_Toc449080442)

[4.2.1文本框（TextBox） 15](#_Toc449080443)

[4.2.2可编辑文本框(EditBox) 16](#_Toc449080444)

[4.2.3按钮(Button) 18](#_Toc449080445)

[4.2.4图片（PictureBox） 19](#_Toc449080446)

[4.2.5菜单列表(MenuList) 19](#_Toc449080447)

[4.2.6签名板（SignatureBoard) 21](#_Toc449080448)

[4.2.7跑马灯（Marquee） 22](#_Toc449080449)

[4.2.8复选框（CheckBox） 22](#_Toc449080450)

[4.2.9单选框（RadioButton） 23](#_Toc449080451)

[4.2.10进度条（ProcessBar） 24](#_Toc449080452)

[4.2.11网格（GridView） 24](#_Toc449080453)

[5 UI库API接口定义 26](#_Toc449080454)

[5.1UI初始化 27](#_Toc449080455)

[5.2加载页面 27](#_Toc449080456)

[5.3设置页面元素属性 28](#_Toc449080457)

[5.4获取页面元素属性 28](#_Toc449080458)

[5.5获取当前页的某个行为 29](#_Toc449080459)

[5.6显示页面并获取页面事件结果 30](#_Toc449080460)

[5.7卸载页 31](#_Toc449080461)

[5.8释放页面资源 32](#_Toc449080462)

[5.9挂起页面 32](#_Toc449080463)

[5.10唤醒页面 32](#_Toc449080464)

[6 使用示例 33](#_Toc449080465)

[7 设计案例 35](#_Toc449080466)

# 1 概述

## 目的

本文档是为了介绍本UI的设计思路，使用方法以及能够支持的界面效果。为有意向使用本UI的用户提供一个操作手册。

## 对象和范围

对象：项目的相关成员，使用本UI框架实现应用界面的用户

范围：本文档适用于在Prolin平台开发界面的用户

## 适用平台

本模块适用于Prolin平台。

## 书写约定

*斜体* 注释性文字

**黑体** 重要信息

**红色** 控件与页相关重要信息说明

**绿色**  字体描述为用户可设置属性

**紫色**  字体描述为用户可获取属性

## 术语

UI 用户界面（User Interface）

API 应用编程接口（Application Program Interface）

OS 操作系统（Operating System）

界面 某一时刻在终端中显示内容，由页组成

页（Page） 屏幕上一块可以绘制文本、图片、按钮等控件的区域，该区域大小可以设置但不能超过屏幕的大小。多页即多个页面同时呈现在屏幕上，但最多同时呈现四个页，即一个状态栏页，一个非接灯页，两个普通页，它们为层叠关系。

控件 控件是具有图形界面的对象如文本框、按钮、图片等

相关关系 xml文件由Page对应的xml脚本组成。其中Page对应的xml脚本由多种控件及其属性组成，页面由Page对应的xml脚本来描述。

# 2简述

为了解决用户对UI的多样需求，设计了一套支持可见可得，开发难度低，界面样式丰富的UI框架。本UI包括UI工具，UI库，XML文件。

## 2.1 EUI设计思想

对于用户来说，如果界面的修改与更换不用跟代码相关联，UI使用起来会更加方便，并且UI设计人员可以不用关心代码流程，更加明确分工。

为了达成设计界面时尽量不需修改代码，代码与UI设计互相独立。EUI提出了一种可见可得，界面以xml脚本来描述，更加容易操作的的一种UI设计。真正做到了修改任何界面，不用动到一行代码。非专业人员经过简单的培训即可在Prolin上做UI的修改或开发。在使用过程中，只需要使用UI库提供的接口显示xml脚本描述的页面即可。

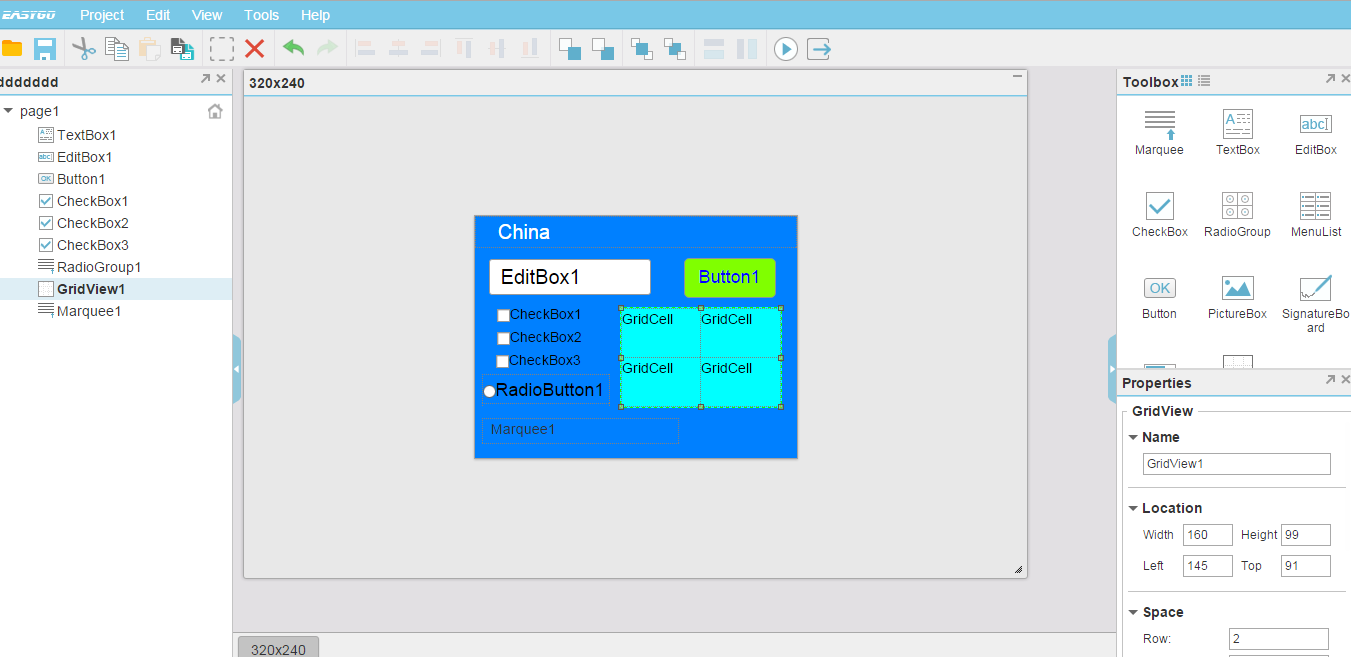
# 3 EUI解决方案的组成

本UI的使用需要搭配三个模块来使用：

1. UI工具，主要是用来设计界面，并以xml格式保存。使用该工具可以先将编辑好的界面模拟出来，配置好后再在终端中呈现，提高开发效率。*（可见可得）*
2. xml文件，由UI工具生成。该文件主要描述所有页面的组成与各控件的属性。*（代码与界面设计相互独立）*
3. UI库，管理输入输出设备，解析xml文件并向用户提供接口。*（界面样式多，开发难度低。）*

## 3.1 UI工具

UI工具需搭配本UI库来使用，这样即可达到界面设计可见可得的目的。UI工具的编辑页面如下：



UI工具可以与MFC一般，以拖拽控件的形式设计页面，并且能设置各个控件的属性。一个工程有且仅有一个xml文件，每个page为xml文件中的一段以<page></page>定义的xml脚本。

工具也能将编辑后的页面模拟出来。如下：

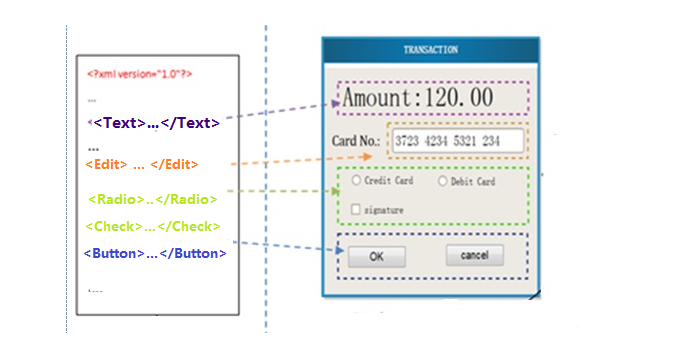


当编辑完成后，工具可以生成该工程所有页面集合的一个xml文件。UI库依靠该xml文件，加载页面，在终端中呈现。

## xml文件

页面，跟平常在浏览器上看到的网页很类似，上面画有文本、图片、按钮，不过它是一个轻量级的页面，页面上可显示的内容在制作页面的工具上会约束，页面由xml脚本描述。

综合考虑发挥当前机器硬件配置下UI的最佳表现效果和UI在未来不同机器的兼容性问题， 我们提出了“一个xml脚本对应一个页面“这里所说的页面，指的是一个对用户可见的和该相关联的所有消息处理的集合，但是页面包括的内容简洁明了，下图为页面控件描述与界面的对应示例：



一个xml对应一个页面，一个页面同一时刻只能呈现一个xml脚本描述的内容，对一个页面加载一个新的xml脚本，必然会导致页面内所有资源的更替。一个xml对应一个页面更想强调的是，应用页面定制化和迁移更便捷，对特定页面的修改一般只需对特定一个xml脚本做替换。这样，即可做到代码与界面设计相互独立。

## UI库

UI库为用户提供了8个接口供使用，具体会在第4节详细介绍。该小节主要介绍UI库的组成框架，对于用户来说只需了解即可。

### UI库组成

UI库主要由以下几个模块来组成：

1. 字库管理，该模块主要是对显示的文字进行加工处理，如调整字体大小，粗斜体，字体类型等属性。系统为字库文件开辟了一个公有空间，这样系统中的应用都可以调到该字库文件。

2. 设备管理器，该模块主要管理与UI相关的设备，如内外置设备的显示，按键，触摸等。

3. 页管理器，解析xml,并存到一个页的结构体中

4. GDI，通过页结构的数据绘制界面。

### UI的使用限制

#### 页数量的限制

本UI支持一个界面最多有四个页面的显示，包括状态栏，非接灯，和两个用户自己配置的页面，我们称之为普通页。

1. 状态栏页：用户只需要使用EUI\_Load接口读取状态栏页的描述名称，即可直接调起状态栏，该页是一个特殊的页，不可被卸载，调起后会一直在应用中存在。UI工具在导出xml文件时会自动导出相应屏幕大小的状态栏xml脚本（该页名不可修改）。

状态栏脚本描述如下：

<Page name="SYS\_STATUSBAR.xml" title="home" x="0" y="0" layoutWidth="240" layoutHeight="20" bgColor="#000000">

<TextBox name="StatusbarTextTime" x="80" y="0" layoutWidth="80" layoutHeight="20" fgColor="#FFFFFF" fontSize="18" textAlign="CENTER\_CENTER"></TextBox>

</Page>

需要注意的是，状态栏已经在UI库中实现，用户只需要调起EUI\_Load("SYS\_STATUSBAR.xml"),即可显示状态栏页，状态栏页的图标文件名是固定的，具体参见EUI demo的res文件夹下的图标资源文件名。

1. 非接灯页：调起方法与状态栏页一样，该页也是特殊页，调起后会在应用中存在。

同样工具会自动导出非接灯页面的xml脚本（该页名不可修改）

非接灯页面xml脚本如下：

<Page name="SYS\_PICCLIGHT.xml" title="home" x="0" y="0" layoutWidth="240" layoutHeight="40" bgColor="#000000" transparent="0">

<PictureBox name="SYS\_PICC\_LIGHT\_BG" x="0" y="0" layoutWidth="240" layoutHeight="40" bgColor="#00000000" transparent="0" imageAlign="CENTER\_CENTER" imagePath=""></PictureBox>

<PictureBox name="SYS\_PICC\_BLUE\_LIGHT" x="0" y="0" layoutWidth="60" layoutHeight="40" bgColor="#00000000" transparent="0" imageAlign="CENTER\_CENTER" imagePath=""></PictureBox>

<PictureBox name="SYS\_PICC\_YELLOW\_LIGHT" x="60" y="0" layoutWidth="60" layoutHeight="40" bgColor="#00000000" transparent="0" imageAlign="CENTER\_CENTER" imagePath=""></PictureBox>

<PictureBox name="SYS\_PICC\_GREEN\_LIGHT" x="120" y="0" layoutWidth="60" layoutHeight="40" bgColor="#00000000" transparent="0" imageAlign="CENTER\_CENTER" imagePath=""></PictureBox>

<PictureBox name="SYS\_PICC\_RED\_LIGHT" x="180" y="0" layoutWidth="60" layoutHeight="40" bgColor="#00000000" transparent="0" imageAlign="CENTER\_CENTER" imagePath=""></PictureBox>

</Page>

非接灯页的资源与状态栏一样，拥有固定的名称，与demo中res文件夹下PICC灯资源文件名一致

1. 普通页：目前最多支持两个普通页在同一个界面中出现，如果普通页的数量超出两个，EUI\_Load将会失败，返回页空间用完（-2）。
2. 虚拟键盘页，由EUI自己调起，用户不需关注，详情见与该文档共同提供的ui.xml文件

#### 控件的限制

对于物理键盘能够获取到焦点的控件（目前只有菜单与编辑框），一个页中只能显示一个，同样一个页中也只能同时显示一个编辑框或菜单。菜单跟编辑框都能够响应物理按键，当页中同时存在菜单与编辑框时，如果有物理按键按下，则页面中是编辑框控件做出响应还是菜单控件做出响应，无法判断（页中同时存在多个编辑框与菜单的情况类似）。

每个页中最多能支持50个控件，包括MenuItem。

# 4 页与控件

本章节将介绍xml文件中页面与各控件和各控件属性的描述规则。如无特殊说明，属性值的类型均为字符串的形式。

## 4.1页

以下是页的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| x | 页相对屏幕（0，0）的横坐标,取值范围（0-65535）如取值不在此范围内，默认为：0 |
| y | 页相对屏幕（0，0）的纵坐标,取值范围（0-65535）如取值不在此范围内，默认为：0 |
| layoutWidth | 页的宽度。取值范围（0-65535）如取值不在此范围内，默认为：0 |
| layoutHeight | 页的高度。取值范围（0-65535）如取值不在此范围内，默认为：0 |
| bgColor | 页的背景颜色 取值范围（0-0xFFFFFFFF，以16进制数表示，如取值不在此范围内，默认为：#00000000） |
| transparent | 页背景透明度 取值范围（0-255，默认为不透明 255）如取值不在此范围内，默认为：255 |
| name | 页控件名称，取值范围（1-252个字符），如取值不在此范围内EUI\_Load返回UI\_RET\_INVALID\_PAGE  必须以.xml结尾，该属性值在screen里必须唯一，以下名称UI库内部使用，所以不可被重名:  SYS\_STATUSBAR.xml、  SYS\_PICCLIGHT.xml、  SYS\_IME\_LAYOUT\_HW.xml、SYS\_IME\_LAYOUT\_EN\_NOCAP.xml、SYS\_IME\_LAYOUT\_EN\_CAP.xml、SYS\_IME\_LAYOUT\_QW.xml、SYS\_IME\_LAYOUT\_PY.xml |

页对应的xml脚本表示的应用示例：

|  |
| --- |
| <Page name=”getcard.xml” title="home" x="0" y="20" layoutWidth="320" layoutHeight="200" bgColor="#FFFFFF">  </Page> |

## 4.2控件

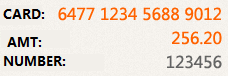
（**绿色字体描述为用户可设置属性，紫色字体为可获取属性，如无特殊说明，所有的控件属性定义均为字符串**）

### 4.2.1文本框（TextBox）

文本框控件是静态类控件的其中一种，该控件不接收和响应输入系统传递的消息。以下是文本框控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识。 |
| value | 静态文本框的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点。 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度 |
| layoutHeight | 控件的高度。 |
| fgColor | 文本框控件显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| textAlign | 文本字符在控件中的对齐方式，取值为：  ( 左上－"TOP\_LEFT"、 中上－"TOP\_CENTER"、  右上－"TOP\_RIGHT"、 左中－"CENTER\_LEFT"、  居中－"CENTER\_CENTER"、右中－"CENTER\_RIGHT"、  左下－"BOTTOM\_LEFT"、 中下－"BOTTOM\_CENTER"、 右下－"BOTTOM\_RIGHT"）  如取值不在此范围内，将默认为CENTER\_CENTER。 |

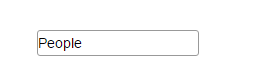
显示示例：



### 4.2.2可编辑文本框(EditBox)

与静态类控件不同，可编缉文本框可接收输入系统传递的消息，通过可编辑文本框，可动态交互式地添加与删减文本内容，所以可编辑文本框也是UI中获取用户信息的重要途径之一。以下是可编辑文本框控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点  当type为 PINPAD，“\*”号时，不能设置该值，否则返回EUI\_ERR\_DATA\_ERR  当type为浮点数时号时，设置该值会无效，但返回成功 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度 |
| layoutHeight | 控件的高度。 |
| fgColor | 显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| textAlign | 字符在控件中的对齐方式，取值为：  左中－"CENTER\_LEFT"  居中－"CENTER\_CENTER"  右中－"CENTER\_RIGHT"  如取值不在此范围内，将默认为CENTER\_CENTER。  Multiline=1时或type=6时，该值设置无效，这时恒为CENTER\_LEFT |
| type | 编辑框接收输入的字符类型**（**0－数字、1－字符串、2－星号密码式、3－浮点数、 4－IP地址、 5－日期、6－PinPad、7—中文输入法**）**  如取值不在该范围内，将默认为0－数字。  type = 5时, format属性不可为空，且value的字符长度必须<=14,否则将返回EUI\_ERR\_INVALID\_PAGE |
| decimal | 当编辑框接收输入的字符类型为小数时，该属性指示小数点精度。其他字符类型，该属性无效（默认：2），取值范围（0-10）。  如取值不在此范围内，EUI\_Load返回UI\_RET\_INVALID\_PARAM |
| minLen | 编辑框最少该接受的字符个数 |
| maxLen | 编辑框目前支持可接受的字符个数  如果maxLen的宽度大于layoutWidth,字符会自动往左边移动。  如maxLen< minLen，EUI\_Load返回EUI\_ERR\_INVALID\_PARAM |
| format | 日期格式规定，type=5时该属性有效.  日期的格式可以用户自己定义，如  Y-M-D h:m:s  y-M-D h:m:s  M-D h:m:s  Y\*&&&^^\*M&&^%D\*&%。  即格式为：YxxMxxDxxhxxmxxs.(xx为不包含以下字符的字符串)。  字段表示：  Y:年份(4位)  y:年份(后2位)  M:月份  D:日期  h:时  m:分  s:秒   1. 如果value值不符合日期格式(如13月32号)，将返回EUI\_ERR\_DATA\_ERR |
| enterReturn | 是否空跳过（TURE:空跳过；FALSE:空不跳过）（如取值不在范围内，默认为：FALSE）  当minLen = maxLen = 0时，该值为TRUE处理，设置无效。 |
| multiline | 是否换行显示，（如取值不在范围内，默认为：1）  0：不换行  1：换行 |
| keyboardType | 0:：物理键盘  1：物理键盘与虚拟键盘混合。当keyboardType = 1时，type，decimal，dateFormat属性设置无效。且物理键盘只能输入数字，无法使用字母键输入字母。（目前只支持240\*320屏幕） |

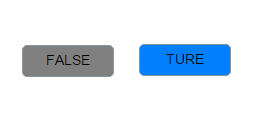


### 4.2.3按钮(Button)

以下是按钮控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| fgColor | 显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| imageAlign | 按钮控件内部图标的位置，取值范围与textAlign一致 |
| displayPattern | 封面图片显示方式，可为  0：普通  1：四角拉伸  如取值不在此范围内，将默认为0 |
| key | 按钮被触发后输出的键值 |
| imagePressed、  imageReleased | 按键被压下时按钮封面装载的图片路径  按键弹起时按钮封面装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |
| enabled | 是否触摸响应  取值为：  TRUE 触摸可响应  FLASE 触摸不响应  如不在取值范围内，默认为TRUE |
| imageDisabled | 按钮失效时装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |

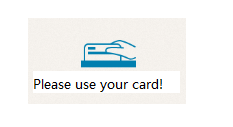
显示示例：



### 4.2.4图片（PictureBox）

与静态文本一样，图片控件不接收和响应输入系统传递的消息，图片可以用于显示BMP、PNG、GIF格式图片。以下是图片控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| bgColor | 图片背景的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| imagePath | 图片资源路径，如路径中没有该图片，将以无图片的形式存在 |
| imageAlign | 控件内部图标的位置，取值范围与align一致 |



### 4.2.5菜单列表(MenuList)

菜单列表控件是一个菜单项容器。

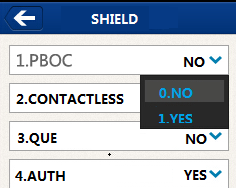
以下是菜单项(MenuItem)控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| itemCaption | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| itemID | 菜单项在菜单列表中的ID号 |
| imagePressed | 按键被压下时按钮封面装载的图片路径，如路径中无图片，将显示无图片 |
| imageReleased | 按键弹起时按钮封面装载的图片路径。如路径中无图片，将显示无图片 |
| Indication | 菜单项标识，通过 EUI\_GetAttribute可以获取到该值，用户可以通过该值进行下一步操作 |
| Enabled | 是否触摸响应  取值为：  TRUE 触摸可响应  FLASE 触摸不响应  如不在取值范围内，默认为TRUE |
| tip | 菜单项的标签内容(支持图片/文字)。均在菜单项右侧显示。  Menulist中tipType=”0”时：tip取值为字符串，字体大小与fontSize大小相同  tipType=”1”时：tip取值为图片路径。如路径中无图片，将返回错误 |

以下是菜单列表控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度 |
| layoutHeight | 控件的高度 |
| fgColor | 菜单项文字的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| scrollScale | 按上下键单次翻阅菜单的个数,取值范围(0---菜单项数)，如超出菜单项数，将默认为0 |
| tipType | 标签类型（文字：0 图片：1），如取值不在范围内，将默认为1 |
| tipColor | 菜单标签文字颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000") |
| lineHeight | 列间距（像素点） |
| lineSpace | 行间距（像素点） |
| numRows | 菜单行数布局（菜单项实际个数超过行列布局总数时，需要翻页显示，显示时，须按照原布局要求显示）默认为：1 |
| numColumns | 菜单列数，默认为：1 |
| textAlign | 字符在控件中的对齐方式，取值为：  ( 左上－"TOP\_LEFT"、 中上－"TOP\_CENTER"、  右上－"TOP\_RIGHT"、 左中－"CENTER\_LEFT"、  居中－"CENTER\_CENTER"、右中－"CENTER\_RIGHT"、  左下－"BOTTOM\_LEFT"、 中下－"BOTTOM\_CENTER"、 右下－"BOTTOM\_RIGHT"）  如取值不在此范围内，将默认为CENTER\_CENTER。只有菜单列数numColumns=1时，该值设置有效 |

显示示例：



### 4.2.6签名板（SignatureBoard)

签名板控件主要用于完成绘制或可选择性地输出用户在触摸屏上划过的轨迹。以下是签名板控件的属性和行为：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度 |
| layoutHeight | 控件的高度 |
| bgColor | 背景色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| SnapShotName | 生成的文件名 |
| brushColor | 画笔颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| brushWidth | 画笔宽度：  1:粗  2:正常  3:细  如不在取值范围内，将默认为2 |
| outputEnabled | 是否将手写轨迹输出（TRUE| FALSE，如不在取值范围内默认为：FALSE） |
| outputDelayTime | 表示当outputEnabled为TRUE时，从用户抬笔到最终坐标点上送的时间间隔，该值为整型。当outputEnabled为FALSE时，该字段不起作用。单位:100ms，取值:10-100 |

### 4.2.7跑马灯（Marquee）

跑马灯主要应用场景可以是以移动的形式显示广告语，一下是跑马灯控件的属性与行为：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| speed | 单位100ms/字符，默认3,即300ms移动一个字符。取值范围为（0—20）,如不在取值范围内，将以0对待 |
| fgColor | 控件显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |

### 4.2.8复选框（CheckBox）

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| fgColor | 显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| imageAlign | 复选框控件内部图标的位置 |
| textAlign | 复选框控件文本的位置 |
| displayPattern | 封面图片显示方式，可为  0：普通  1：四角拉伸  如取值不在此范围内，将默认为0 |
| key | 复选框被触发后输出的键值 |
| imagePressed、  imageReleased | 复选框被压下时按钮封面装载的图片路径  复选框弹起时按钮封面装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |
| enabled | 是否触摸响应  取值为：  TRUE 触摸可响应  FLASE 触摸不响应  如不在取值范围内，默认为TRUE |
| checked | 是否选中  TRUE 选中  FLASE 不选中  如不在取值范围内，默认为FLASE |
| imageDisabled | 复选框失效时装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |

### 4.2.9单选框（RadioButton）

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| fgColor | 显示文本的颜色，取值范围（0-FFFFFFFF，以16进制数表示。如取值不在此范围内，默认为："#00000000"，以后六位为准。前两位FF为扩展用，目前无效。) |
| fontSize | 字符的大小，取值范围（5-100）。  如取值不在此范围内，fontSize默认为5  如取值fontSize > layoutHeight，将显示不出文字。 |
| imageAlign | 单选框控件内部图标的位置 |
| textAlign | 单选框控件文本的位置 |
| displayPattern | 封面图片显示方式，可为  0：普通  1：四角拉伸  如取值不在此范围内，将默认为0 |
| key | 单选框被触发后输出的键值 |
| imagePressed、  imageReleased | 单选框被压下时按钮封面装载的图片路径  单选框弹起时按钮封面装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |
| enabled | 是否触摸响应  取值为：  TRUE 触摸可响应  FLASE 触摸不响应  如不在取值范围内，默认为TRUE |
| checked | 是否选中  TRUE 选中  FLASE 不选中  如不在取值范围内，默认为FLASE |
| imageDisabled | 单选框失效时装载的图片路径  如路径中图片不存在，将以无图片的形式显示。 |

### 4.2.10进度条（ProcessBar）

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x | 控件相对页的横坐标 |
| y | 控件相对页的纵坐标 |
| value | 进度值（大于等于“0” & 小于等于“100”） |
| layoutWidth | 控件的宽度。 |
| layoutHeight | 控件的高度。 |
| imageBgUrl  imageFgUrl | 背景图片路径  前景图片路径 |

### 4.2.11网格（GridView）

GridView是GridCell的容器

以下是网格项(GridCell)控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| value | 静态文本的显示内容，如果字符串的宽度大于控件宽度将超出的部分显示三个黑点 |
| startRow  endRow  startColumn  endColumn | 该网格项控件所占的行数与列数（由0开始） |
| bgColor | 背景色（"#RRGGBB") |
| fgColor | 字体颜色（"#RRGGBB") |
| fontSize | 字体大小 |
| textAlign | 字符对齐方式 |
| cellWidth | 当前单元格的宽度 |

以下是网格(GridView)控件的属性：

|  |  |
| --- | --- |
| **属性名称** | **属性说明** |
| name | 控件的名称，是控件在页中的唯一标识 |
| x, y | 分别为控件相对页的横、纵坐标 |
| layoutWidth | 控件的宽度 |
| layoutHeight | 控件的高度 |
| bgColor | 背景色（"#RRGGBB") |
| scrollScale | 滚动刻度（1—100%） |
| numRows | 行数 |
| numColumns | 列数 |

# 5 UI库API接口定义

只用10个接口，即可实现大部分界面效果，并支持彩屏，各类大小的屏幕，支持滑动，半透明，毛玻璃的界面效果。接口容易理解，开发难度低。

|  |
| --- |
| **返回值宏定义：**  #define EUI\_RET\_OK 0  enum PDK\_EUI\_RET //scanner module(-3100~-3199)  {  EUI\_ERR\_BASE =(0-3100),  EUI\_ERR\_INVALID\_PARAM =(EUI\_ERR\_BASE-1),  EUI\_ERR\_MEM\_RUN\_OUT =(EUI\_ERR\_BASE-2),  EUI\_ERR\_INVALID\_TAG =(EUI\_ERR\_BASE-3),  EUI\_ERR\_OPER\_ERR =(EUI\_ERR\_BASE-4),  EUI\_ERR\_TIME\_OUT =(EUI\_ERR\_BASE-5),  EUI\_ERR\_NO\_RESULT =(EUI\_ERR\_BASE-6),  EUI\_ERR\_INVALID\_CONFIG =(EUI\_ERR\_BASE-7),  EUI\_ERR\_INVALID\_CHANNEL =(EUI\_ERR\_BASE-8),  EUI\_ERR\_INVALID\_PAGE =(EUI\_ERR\_BASE-9),  EUI\_ERR\_INITIALIZED\_FAILED =(EUI\_ERR\_BASE-10),  EUI\_ERR\_ERR\_STATE =(EUI\_ERR\_BASE-11),  EUI\_ERR\_MSG\_TIMEOUT =(EUI\_ERR\_BASE-12),  EUI\_ERR\_MSG\_ERR =(EUI\_ERR\_BASE-13),  EUI\_ERR\_DATA\_ERR =(EUI\_ERR\_BASE-14),  EUI\_ERR\_FONT\_NO\_FREE =(EUI\_ERR\_BASE-15),  EUI\_ERR\_FONT\_INIT\_FALIED =(EUI\_ERR\_BASE-16),  EUI\_ERR\_XML\_NO\_FREE =(EUI\_ERR\_BASE-17),  EUI\_ERR\_XML\_INIT\_ERR =(EUI\_ERR\_BASE-18),  EUI\_ERR\_PAGEMANAGER\_INIT\_ERR=(EUI\_ERR\_BASE-19),  }; |

## 5.1UI初始化

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Init(char \*pszFontFilePath, char \* pszXmlFilePath);** | | |
| 函数功能 | UI初始化 | | |
| 参数说明 | pszFontFilePath  **[输入]** | 字库路径（NULL为不加载） | |
| pszXmlFilePath  **[输入]** | Xml文件路径（NULL为不加载） | |
| 返回值 | 见返回值定义 | |  |
| 用　法 | 在第一次使用UI前必须字库与XML文件都不为空 | | |
| 备 注 | 通过EUI\_Init来初始化状态栏，字库，页管理器等模块。字库与XML文件路径需在MAINAPP。EUI\_Init与EUI\_Free成对使用，如果不释放某一种资源，并且再次加载该资源，会返回该资源没有释放的错误。 | | |

## 5.2加载页面

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Load(uint channel, char \*PageName)** | | |
| 函数功能 | 解析xml为**当前**预显示的页 | | |
| 参数说明 | channel **[输入]** | 默认为0，该参数为扩展参数 | |
| PageName **[输入］** | 描述页的xml脚本名称 (必须以".xml"结尾） | |
| 返回值 | EUI\_RET\_OK | | 加载成功 |
| UI\_RET\_INITIALIZED\_FAILED | | 没有初始化 |
| UI\_RET\_INVALID\_PARAM | | 入参错误 |
| UI\_RET\_INVALID\_CONFIG | | 配置无效 |
| UI\_RET\_MEM\_RUN\_OUT | | 内存溢出（页面超限） |
| UI\_RET\_ERR\_STATE | | 状态错误 |
| 用　法 | 加载一个新的页：  　　　　．．．  **EUI\_Load(0, "payment.xml");** | | |
| 备 注 | 1. 加载后的页被置为当前聚焦的页，即最上层的页，简称**当前页，**非当前页不能接受事件。  2. 调用该接口加载页后，页内容在调用EUI\_Process后更新。 | | |

## 5.3设置页面元素属性

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_SetAttribute(uint channel, char \*name, const char \*attr, char \*value)** | | |
| 函数功能 | 设置**当前**页内控件属性 | | |
| 参数说明 | channel **[输入］** | 默认为0，该参数为扩展参数 | |
| name **[输入］** | 控件名称 | |
| attr **[输入]** | 控件属性名称 | |
| value **[输入]** | 控件属性的值 | |
| 返回值 | EUI\_RET\_OK | | 设置成功 |
| <0 | | 见返回值定义 |
| 用　法 | 加载一个页面，并且修改当前页面上一个编辑框("amount")的最大可接受内容长度为５个字符：  　　．．．  EUI\_Load(0, "payment.xml");  **EUI\_SetAttribute(**0**, "amount", "maxLen", "5");** | | |
| 补充说明 | 1.页面元素名称在页中必须唯一。  2. 设置页面控件属性后，页面内容在调用EUI\_Process后更新。  3. 对于可设置的页面控件属性UI做了一些限制，即有部分属性是不可设置的，在第4节已详细介绍哪些属性有可设置与可获取的权限。  4. 如未加载页面直接调用该接口，将返回-9.  5. 对于编辑框控件的value属性，如果设置的value属性值与type不相符，将返回EUI\_ERR\_DATA\_ERR。  6. 除了文本框的value属性，其他控件均不可多次调用该函数来修改控件属性值。 | | |

## 5.4获取页面元素属性

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_GetAttribute(uint channel, char \*name, char \*attr, char \*value)** | | |
| 函数功能 | 获取**当前**页内控件属性 | | |
| 参数说明 | channel **[输入］** | 默认为0，该参数为扩展参数 | |
| name **[输入］** | 控件名称 | |
| attr **[输入]** | 控件属性名称 | |
| value **[输出]** | 控件属性的值 | |
| 返回值 | EUI\_RET\_OK | | 获取成功 |
| <0 | | 见返回值定义 |
| 用　法 | 加载一个页面，并且获取当前页面上一个编辑框("amount")的最大可接受内容长度　　．．．  EUI\_Load(0, "payment.xml");  **EUI\_GetAttribute(**0**, "amount", "maxLen", szOut);** | | |
| 补充说明 | 1. 页面元素名称在页中必须唯一。 2. 如未加载页面直接调用该接口，将返回-9. | | |

## 5.5获取当前页的某个行为

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_RequestAction(uint channel, char \*name, char \*action, char \*value)** | | |
| 函数功能 | 获取当前页的某个行为 | | |
| 参数说明 | channel **[输入］** | 默认为0，该参数为扩展参数 | |
| name **[输入］** | 控件名称  目前只支持签名板控件 | |
| action **[输入]** | 控件行为  目前支持的属性有  签名板的：SnapScreen(value为true时，截取当前屏幕所有内容)  ClearSign(value为true时，清空当前签名板的签名轨迹) | |
| value **[输入]** | 控件属性的值 | |
| 返回值 | EUI\_RET\_OK | | 获取成功 |
| <0 | | 见返回值定义 |
| 用　法 | 截取当前签名板的签名轨迹，并生成一个名为sign\_file\_name.bmp的图片  EUI\_Load(0, "sign.xml");  EUI\_SetAttribute(0, "sign", "SnapShotName", “sign\_file\_name.bmp”);  EUI\_Process(0,TIMEOUT,szOutData,size,sizeof(szOutData));  EUI\_RequestAction(0, "sign", "SnapScreen", "true"); | | |
| 补充说明 |  | | |

## 5.6显示页面并获取页面事件结果

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Process(uint channel, int timeout, uchar \*dataOut, uint size, int \*lenOut,)** | | |
| 函数功能 | 显示所有已加载的页并获取**当前**页面事件响应的结果 | | |
|  | channel **[输入]** | 默认为0，该参数为扩展参数 | |
| dataOut **[输出］** | 执行成功后返回的数据 | |
| size 　　 **[输入]** | 传入数据缓冲区的大小 | |
| lenOut 　 **[输出]** | 输出数据的长度 | |
| timeout **[输入]** | 超时参数，单位为秒  <0：无限等待时间  >=0：等待timeout(s) | |
| 返回值 | <0 | | 见返回值定义 |
| EUI\_RET\_OK | | 执行成功，显示超时退出 |
| 1 | | 该页面有返回值，出参dataOut为逻辑键值。 |
| 2 | | 该页面有返回值，出参dataOut为字符串。 |
| 3 | | 该页面有返回值，出参dataOut为页面中菜单项MenuItem的name定义 |
| 4 | | 该页面有返回值，出参dataOut为签名板的点阵轨迹 |
| 7 | | 该页面有返回值，出参为多编辑框的TLV值， |
| 用　法 | 加载一个页面，并等待用户触发页面事件：  **．．．**  　EUI\_Load(0 "payment.xml");  while(1)  {  **ret = EUI\_Process(**0**, 0, buffer, sizeof(buffer), &** **lenOut );**  if(ret > 0) break;  } | | |
| 补充说明 | 对于所有控件，确认键与取消键均可响应，返回值为1，dataOut出参为按下的确认键或取消键键值  对于除编辑框与菜单之外的控件， 按下任何按键，返回值均为1，dataOut为按下的键值。  对于带编辑框的页面，如有输入值时，将返回2，dataOut为输入的字符串。  对于带菜单的页面，如有1-9按键，将返回3，dataOut为按键对应菜单项的name值。  对于带签名板的页面，且支持轨迹输出，将返回4，dataOut为轨迹的点阵数据。  对于有两个以上编辑框的页面，输入完毕之后按确认键，将返回7，dataOut为这些编辑框输入数据的组合。格式为：  编辑框个数（1字节）+第0个编辑框编号（1字节）+第0个编辑框数据长度（1字节）+第0个编辑框数据（N字节）+第1个编辑框编号+第1个编辑框数据长度（1字节）+第1个编辑框数据（N字节）……..依次类推 | | |

## 5.7卸载页

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Flush(int channel)** | | |
| 函数功能 | 卸载当前页 | | |
| 参　数 | channel **[输入]** | . 默认为0，该参数为扩展参数 | |
| 返回值 | EUI\_RET\_OK | | 执行成功 |
| <0 | | 见返回值定义 |
| 用　法 | 加载一个页面，等待一个页面响应事件，页面事件被触发后，获取页面响应的结果，之后卸载页。  ．．．  　EUI\_Load(0 "payment.xml");  while(1)  {  ret = EUI\_Process(0, 0, buffer, sizeof(buffer), & lenOut );  if(ret > 0) break;  　}  **EUI\_Flush (0);** | | |
| 补充说明 | 1. 页面被卸载后, 页面内容在调用EUI\_Process后更新。  2. 当前页被卸载后，原处于下一层的页被置为当前页。（多页情况） | | |

## 5.8释放页面资源

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Free(int iFontFree,int iXmlFree)** | | |
| 函数功能 | 释放页面资源 | | |
| 参　数 | **iFontFree** **[输入]** | 0：不释放字库文件  非0：释放字库文件 | |
| **iXmlFree [输入]** | 0：不释放XML文件  非0：释放XML文件 | |
| 返回值 | EUI\_RET\_OK | | 执行成功 |
| <0 | | 见返回值定义 |
| 用　法 |  | | |
| 补充说明 | 调用该函数时，字库与XML文件都可以选择是否释放，当释放任意一种文件后，再次使用时必须调用EUI\_Init重新加载，反之则不需要重新加载。 | | |

## 5.9挂起页面

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Suspend(int channel)** | | |
| 函数功能 | 挂起当前页面 | | |
| 参　数 | channel **[输入]** | 默认为0，该参数为扩展参数 | |
| 返回值 | UI\_RET\_OK | | 执行成功 |
| <0 | | 见返回值定义 |
| 用　法 | 切换应用时，必须将本应用的当前页面挂起。  ．．．  　Ret = EUI\_Suspend (0); | | |
| 补充说明 |  | | |

## 5.10唤醒页面

|  |  |  |  |
| --- | --- | --- | --- |
| 函数原型 | **int EUI\_Resume(int channel)** | | |
| 函数功能 | 唤醒已挂起的页面 | | |
| 参　数 | channel **[输入]** | 默认为0，该参数为扩展参数 | |
| 返回值 | EUI\_RET\_OK | | 执行成功 |
| <0 | | 见返回值定义 |
| 用　法 | 切换应用时，必须将本应用的当前页面唤醒。  ．．．  　Ret = EUI\_Resume (0); | | |
| 补充说明 |  | | |

# 6 使用示例

widget（name）

Page(home\_screen.xml)

TextBox(week)

TextBox(date)

PictureBox(logo)

TextBox(prompt)

TextBox(version)



如上图所示，在xml中的描述为：

|  |
| --- |
| <Root>  <Screen name="320x240" width="320" height="240">  <!--待机界面-->  <Page name="home\_screen.xml" title="home" x="0" y="20" layoutWidth="320" layoutHeight="220" bgColor="#FFFFFFFF">  <PictureBox name="logo" x="0" y="0" layoutWidth="320" layoutHeight="220" bgColor="#000000" transparent="255" imageAlign="CENTER\_CENTER" imagePath="cup.png"></PictureBox>  <TextBox name="version" x="0" y="110" layoutWidth="320" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_CENTER"></TextBox>  <TextBox name="prompt" x="0" y="130" layoutWidth="320" layoutHeight="30" fgColor="#FFFFFF" fontSize="30" textAlign="CENTER\_CENTER"></TextBox>  <TextBox name="date" x="0" y="168" layoutWidth="160" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_LEFT"></TextBox>  <TextBox name="week" x="140" y="168" layoutWidth="160" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_RIGHT"></TextBox>  </Page>  <Page name="home\_screen1.xml" title="home" x="0" y="20" layoutWidth="320" layoutHeight="220" bgColor="#FFFFFFFF">  <PictureBox name="logo" x="0" y="0" layoutWidth="320" layoutHeight="220" bgColor="#000000" transparent="255" imageAlign="CENTER\_CENTER" imagePath="cup.png"></PictureBox>  <TextBox name="version" x="0" y="110" layoutWidth="320" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_CENTER"></TextBox>  <TextBox name="prompt" x="0" y="130" layoutWidth="320" layoutHeight="30" fgColor="#FFFFFF" fontSize="30" textAlign="CENTER\_CENTER"></TextBox>  <TextBox name="date" x="0" y="168" layoutWidth="160" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_LEFT"></TextBox>  <TextBox name="week" x="140" y="168" layoutWidth="160" layoutHeight="20" fgColor="#FFFFFF" fontSize="20" textAlign="CENTER\_RIGHT"></TextBox>  </Page>  </Screen>  </Root> |

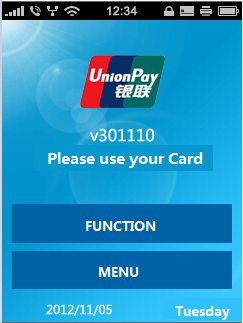
该页的name为home\_screen.xml，假如程序中希望将这个页面在终端中显示，代码示例为：

|  |
| --- |
| int main()  {  ……..  EUI\_Init(0);  iRet = EUI\_Load(0," home\_screen.xml ");  ……..  iRet = EUI\_Process(0,TIMEOUT,OutData,sizeof(OutData),&len);  ……..  EUI\_Flush(0);  } |

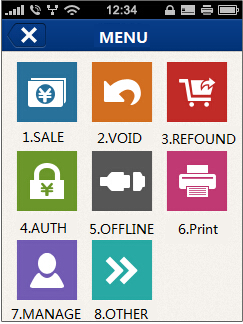
如此，即可将用xml描述好的界面在终端中显示出来。

# 7 设计案例

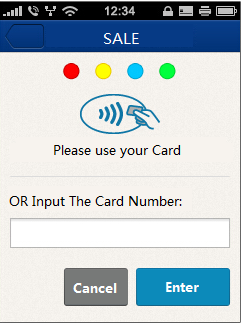
**图片与虚拟按钮：**



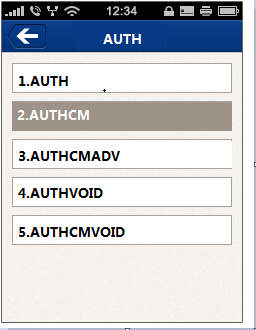
**九宫格菜单虚拟按键：**



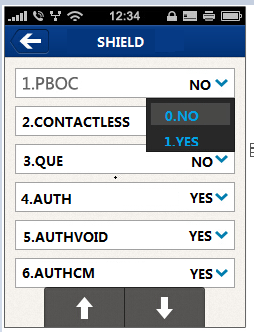
**编辑框与虚拟灯：**



**单列菜单按键效果：**



**双页显示效果：**



**文本框与编辑框：**

