FAW HS7

IVI System Design

AppUI-API User Guide

Released on the: 2016.12.20

Version Number: 0.650

Neusoft Automotives ChangChun

**Version History**

| No. | Version | Section | Brief Description | Date | Rseponsible |
| --- | --- | --- | --- | --- | --- |
| 1 | 0.650 | All | Initial Create | 2016-12-20 | ZhangShuang |
| 2 | 0.651 | 3 | 更新目录，添加3.3 3.5两节的具体内容 | 2016-1-16 | ZhangShuang |
| 3 | 0.652 | 3 | 更新目录，添加3.7 3.8 3.9两节的具体内容，添加3.10节， | 2017-02-13 | ZhangShuang |
| 4 | 0.653 | 3.11 | 为新API添加说明 | 2017-08-23 | ZhangShuang |
| 5 | 0.87 | 3.3、3.5、3.10 | 增加surface大小调整函数的使用限制说明。 | 2017-12-11 | ZhangShuang |
| 6 | 0.92 | 3.10 3.11 | 插入新的3.10 3.11节，原3.10 3.11节及之后章节顺次后延 | 2018-03-13 | ZhangShuang |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Content

[1 概述 4](#_Toc509240322)

[2 文件组成 4](#_Toc509240323)

[3 接口说明 4](#_Toc509240324)

[3.1 AppUI-API的使用约束 4](#_Toc509240325)

[3.2 Surface是如何显示到屏幕上的 4](#_Toc509240326)

[3.3 设置Surface的Source Rectangle 4](#_Toc509240327)

[3.4 获取Surface的Source Rectangle 5](#_Toc509240328)

[3.5 设置Surface的Destination Rectangle 5](#_Toc509240329)

[3.6 获取Surface的Destination Rectangle 5](#_Toc509240330)

[3.7 隐藏当前程序的Titlebar 5](#_Toc509240331)

[3.8 显示当前程序的Titlebar 5](#_Toc509240332)

[3.9 获取当前程序的Titlebar的显示状态 5](#_Toc509240333)

[3.10 请求拉下/收起Titlebar 6](#_Toc509240334)

[3.11 查询当前屏幕Titlebar是拉下还是收起 6](#_Toc509240335)

[3.12 同时设置Surface的Source Rectangle和Destination Rectangle 6](#_Toc509240336)

[3.13 设置Under层surface的对齐方式 6](#_Toc509240337)

[4 Refrences 7](#_Toc509240338)

# 概述

AppUI-API是一组用于调整App外观的API。

# 文件组成

头文件：/usr/include/appSDK/AppSDK.h

库文件：/usr/lib/libappSDK.so

# 接口说明

## AppUI-API的使用约束

约束条件：

* 系统中AppManager进程正常运行；
* 此应用程序通过AppManager启动；
* 应用程序使用传递的surfaceID成功创建surface；
* 使用AppUI-API前需要首先初始化AppSDK；

## Surface是如何显示到屏幕上的

一个surface是ilm管理显示输出的最小单位，可以认为是一块“画布”

这块“画布”显示到屏幕上需要经过两次变换：

1. 对surface需要显示到屏幕上的区域进行标记，标记出来的区域称之为“Source Rectangle”
2. 对surface将要显示到的layer上的区域进行标记，标记出来的区域称之为“Destination Rectangle”

ilm在显示一个surface时，会截取它的source rectangle，然后将source rectangle拉伸至destination rectangle标记的区域。如果这个区域刚好在屏幕可见范围内，那么这个surface就会被显示出来。

## 设置Surface的Source Rectangle

**注意：**这个函数只能在当前APP第一次onpostrender之后使用。在此之前调用会导致全部的surface调整函数失效！

|  |  |  |
| --- | --- | --- |
| *declare* | void SetSourceRectangle(const int& x, const int& y, const unsigned int& width, const unsigned int& height) | |
| *description* | 设置surface的source rectangle | |
| *parameter* |  |  |
| *return* | null | |

## 获取Surface的Source Rectangle

## 设置Surface的Destination Rectangle

**注意：**这个函数只能在当前APP第一次onpostrender之后使用。在此之前调用会导致全部的surface调整函数失效！

|  |  |  |
| --- | --- | --- |
| *declare* | void SetDestinationRectangle (const int& x, const int& y, const unsigned int& width, const unsigned int& height) | |
| *description* | 设置surface的destination rectangle | |
| *parameter* |  |  |
| *return* | null | |

## 获取Surface的Destination Rectangle

## 隐藏当前程序的Titlebar

|  |  |  |
| --- | --- | --- |
| *declare* | void HideTitlebar(int style = 0) | |
| *description* | 隐藏当前程序的Titlebar | |
| *parameter* |  |  |
| *return* | null | |

## 显示当前程序的Titlebar

|  |  |  |
| --- | --- | --- |
| *declare* | void ShowTitlebar(int style = 0) | |
| *description* | 显示当前程序的Titlebar | |
| *parameter* |  |  |
| *return* | null | |

## 获取当前程序的Titlebar的显示状态

|  |  |  |
| --- | --- | --- |
| *declare* | bool IsTitlebarVisible() | |
| *description* | 获取当前程序的Titlebar的显示状态（可见/不可见） | |
| *parameter* |  |  |
| *return* | Bool true表示可见，false表示不可见。 | |

## 请求拉下/收起Titlebar

Titlebar支持通过拖动拉下的方式以展示更多内容。此API的作用就是控制Titlebar下拉。

Titlebar的下拉/收起与[3.8节](#_隐藏当前程序的Titlebar) [3.9节](#_显示当前程序的Titlebar)叙述的Titlebar的显示/隐藏是两个相互独立的状态，**在编写Titlebar App时需要注意**。

|  |  |  |
| --- | --- | --- |
| *declare* | void DropDownTitlebar(bool isDropDown = true, int style = 0) | |
| *description* | 请求拉下/收起Titlebar | |
| *parameter* |  |  |
| *return* | Bool true表示下拉，false表示收起。 | |

style参数目前没有用到。但是Titlebar App能接收到这个值。

## 查询当前屏幕Titlebar是拉下还是收起

|  |  |  |
| --- | --- | --- |
| *declare* | bool IsTitlebarDropDown() | |
| *description* | 查询当前屏幕Titlebar是拉下还是收起 | |
| *parameter* |  |  |
| *return* | Bool true表示下拉，false表示收起。 | |

## 同时设置Surface的Source Rectangle和Destination Rectangle

因为分别设置Source Rectangle和Destination Rectangle可能导致一些显示问题，所以添加了这个函数同时设置两个值。这个函数只适用于source Rectangle 和 destination Rectangle为同一个值的情况。

**注意：**这个函数只能在当前APP第一次onpostrender之后使用。在此之前调用会导致全部的surface调整函数失效！

|  |  |  |
| --- | --- | --- |
| *declare* | void SetBothRectangle(const int& x, const int& y, const unsigned int& width, const unsigned int& height) | |
| *description* | 同时设置当前程序HMI层surface的source Rectangle 和 destination Rectangle为同一个值。 | |
| *parameter* |  |  |
| *return* | Null | |

## 设置Under层surface的对齐方式

因为Under层surface的尺寸可能会比屏幕小，所以SDK提供了这个函数用于调整这个surface的对齐方式。目前只支持居中和靠左上角。（如果surface尺寸大于屏幕，则它只能靠左上角显示。因为wayland不支持负数坐标，而屏幕左上角的坐标是（0,0）。）

Under层surface默认是居中显示的。

|  |  |  |
| --- | --- | --- |
| *declare* | void SetUnderLayerCenter(bool isCenter = true) | |
| *description* | 设置Under层surface是否居中显示 | |
| *parameter* | bool | 默认为true，即居中显示 |
| *return* | Null | |

|  |  |  |
| --- | --- | --- |
| *declare* | bool IsUnderLayerCenter() | |
| *description* | 查询Under层surface是否居中显示 | |
| *parameter* | Null |  |
| *return* | 返回true则是居中显示，否则靠左上角显示 | |

# Refrences