

# V40 Carlet 项目

## 分屏功能配置说明

## 文档履历

[illegible]

# 目 录

目 录 .....	3
1. 引言 .....	4
1.1. 编写目的 .....	4
1.2. 适用范围 .....	4
1.3. 参考人员 .....	4
2. 分屏简介 .....	5
3. 拓展接口说明 .....	6
3.1. ActivityManager 接口 .....	6
3.2. Activity 接口 .....	8
3.3. StatusBarManager 接口 .....	9
3.4. Intent 相关定义 .....	9
4. 下拉菜单配置 .....	10
4.1. 数据存储位置 .....	10
4.2. 配置方法 .....	10
4.3. 注意内容 .....	10
5. 屏幕适配 .....	11
5.1. 分屏窗口大小配置 .....	11
6. 分屏开关配置 .....	12
6.1. 编译时的配置项 .....	12
6.2. 系统运行时的配置项 .....	12
7. 系统默认应用配置 .....	13
7.1. 数据存储位置 .....	13
7.2. 配置方法 .....	13
8. 右窗口应用开发注意事项 .....	14
9. 调试方法 .....	15
9.1. 查看左右窗口任务栈的内容及窗口区域 .....	15
9.2. 查看左右窗口的状态 .....	15
9.3. 打开分屏调试 log .....	16
Declaration .....	17

## 1. 引言

### 1.1. 编写目的

本详细设计说明书目的在于说明分屏功能的设计考虑，包括程序描述，输入/输出，逻辑处理流程等。为软件编程和系统维护提供基础，以利于程序员编制程序。

### 1.2. 适用范围

适用于 Android 4.4 系统。

### 1.3. 参考人员

Android 软件开发人员，系统维护人员，FAE。

## 2. 分屏简介

分屏是将一块屏幕分成多个区域显示。V40 的分屏技术将屏幕分成 ABC 三个显示区域。A 区显示系统信息及系统按键，B 区为主显示区域，用于运行主要的程序，这个区域要确保能运行所有的应用程序，并且该区域对所运行的应用程序没有限制。最右边的 C 区为副显示区域，它用于运行特定的应用程序。



图 1-1 分屏界面

### 3. 拓展接口说明

#### 3.1. ActivityManager 接口

##### 获得左窗口 ActivityStack Id

定义:

```
/** @hide */  
  
public int getLeftStackId()
```

功能: 返回左窗口 Activity Stack Id, 返回值小于 0 是无效的。当返回小于 0 时, 说明主显示区域不存在窗口, 当前显示的是 launcher。

##### 获得右窗口 ActivityStack Id

定义:

```
/** @hide */  
  
public int getRightStackId()
```

功能: 返回右窗口 Activity Stack Id, 返回值小于或等于 0 都是无效的。

##### 获得指定窗口的尺寸状态

定义:

```
/** @hide */  
  
public int getWindowSizeStatus(int stackId)
```

功能: 获得给定 Activity Stack Id 窗口的大小状态, 输入参数可以是左窗口或者右窗口的 Activity Stack Id。返回值表示窗口的状态, 总共有四种状态: 分屏, 全屏, 隐藏, 无效。四种状态分别对应的值是:

表 2-1

窗口状态	常量定义	值
分屏	MW_NORMAL_STACK_WINDOW	0
全屏	MW_MAX_STACK_WINDOW	1
隐藏	MW_MIN_STACK_WINDOW	2
无效	MW_INVALID_STACK_WINDOW	-1

如果传入的 Activity Stack Id 值错误, 或者窗口不存在, 将会返回-1, 在使用返回值之前, 强

烈建议先判断窗口返回值是否有效。

主显示区域中显示的应用和 **launcher** 都具有相同的窗口尺寸状态，如果主显示区域没有运行应用时，可以查询 **launcher** 窗口的状态，同样可以反映出当前屏幕显示的状态。

## 设置窗口的尺寸状态

定义：

```
/** @hide */
public void setWindowSize(int stackId, int size)
```

功能：设置指定 Activity Stack Id 的窗口的状态，设置的状态只能是上述窗口的：分屏、全屏、隐藏三种状态。参数 size 是上述给定的值。

**注意事项：**

1. 获取得到 Activity Stack Id 之后，应当判断其是否有效，然后再使用；
2. 获取得到窗口尺寸状态后，应当判断其是否有效，然后再使用；
3. 当应用从右窗口移动到左窗口时，请使用 2.2 节介绍的 **onActivityMove** 方法做相应处理。

## 获得指定 ActivityStack 顶部的 task

定义：

```
public RecentTaskInfo getTopTaskOfStack(int stackId)
```

参数：stackId 为指定的 ActivityStack 的 stack id，必须大于或等于 0，当为 0 时，获取得到的是 launcher 的信息。返回值为 task 的信息，如果传入的参数无效或者查询不到信息时，返回值为 null。

功能：获得指定 stackId 的 ActivityStack 顶部的 task，配合 **getLeftStackId** 和 **getRightStackId**，可以获得左右窗口顶部的应用信息。

示例：

```
int rightStackId = -1;
int leftStackId = -1;
rightStackId = am.getLeftStackId();
leftStackId = am.getRightStackId();
ComponentName rightComponentInfo = null;
ComponentName leftComponentInfo = null;
ActivityManager.RecentTaskInfo rightTask =
```

```
        am.getTopTask(rightStackId);

        ActivityManager.RecentTaskInfo leftTask =

        am.getTopTask(leftStackId);

        if (rightTask != null) {

            rightComponentInfo = rightTask.baseIntent.getComponent();

        }

        if (leftTask != null) {

            leftComponentInfo = leftTask.baseIntent.getComponent();

        }

    }
```

获取 ActivityStack 位置

定义：

```
public int getStackPostition(int stackId)
```

参数：stackId 为需要查询位置信息的 ActivityStack ID 值，返回值及含义如下：

表 2-2

返回值	含义
-1	无效
0	ActivityStack 的窗口在右分屏
1	ActivityStack 的窗口在左分屏

功能：判断应用所处的位置。

3.2.Activity 接口

应用显示区域切换响应函数

定义：

```
protected void onActivityMove(boolean isToLeft)
```

参数：isToLeft 为 true 时，表示应用从副显示区域移到主显示区域；isToLeft 为 false 时，表示应用从主显示区域移到副显示区域。

功能：当用户操作将应用在两个显示区域移动时，系统会主动调用 onActivityMove 函数，应用开发人员可以重写此函数，以响应切换事件。

用法：在应用的 Activity 中重写 Activity 的 onActivityMove 函数。



```

@Override

protected void onActivityMove(boolean isToLeft) {

    // TODO Auto-generated method stub

    Log.d(TAG, "hcl onActivityMove:"

        + (isToLeft ? "right" : "left") + " -> " + (isToLeft ? "left" : "right"));

    super.onActivityMove(isToLeft);

}

```

### 3.3.StatusBarManager 接口

#### 控制分屏按钮显示与隐藏

定义:

```
public void setSplitButtonVisibility(boolean isVisible) throw RuntimeException(ex)
```

参数: isVisible 为 true 时, 设置分屏按钮可见, isVisible 为 false 时, 设置分屏按钮隐藏。

功能: 用于控制分屏按钮的显示与隐藏。

注意: 滥用此方法, 有可能导致分屏按钮显示异常。应用将分屏按钮隐藏后, 需要在合适的时机, 将分屏按钮恢复显示, 并且需要做好容错机制, 在异常发生后, 及时恢复分屏按钮的显示。

### 3.4.Intent 相关定义

#### 指定应用在右窗口显示

定义:

```
public static final int FLAG_ACTIVITY_RUN_IN_RIGHT_WINDOW = 0x00000200
```

功能: 当通过 intent 打开一个应用时, 希望应用在右窗口显示, 可以通过 Intent.addFlags 添加该标志位。AMS 将会根据此标记位, 在右窗口打开应用。

#### 窗口状态已经变更

定义:

```
public static final String ACTION_SPLIT_WINDOW_HAS_CHANGED = \

    "android.intent.action.SPLIT_WINDOW_HAS_CHANGED"
```

功能: 当窗口由分屏切到全屏, 或者由全屏切到分屏时, 系统将会发出此广播。

## 4. 下拉菜单配置

### 4.1. 数据存储位置

下拉菜单应用列表的配置内容存放在 settings.db 数据库的 system 表中, 字段名是 right\_memu\_app\_list, 数据类型是 String, 有效数据以"--split--"字符串隔离。

### 4.2. 配置方法

在 overlay 中添加应用包名, 具体方法如下: 在 android\device\softwinner\aston-xxx\overlay\frameworks\base\packages\SettingsProvider\res\values\defaults.xml 中添加以下内容:

```
<string name="def_right_memu_app_list" translatable="false">包名 1--split--包名 2...</string>
```

如果在上述目录中, 不存在 defaults.xml 文件, 则新建一个 default.xml 文件, 并加入以下内容:

```
<?xml version="1.0" encoding="utf-8"?>

<resources>

    <string name="def_right_memu_app_list" translatable="false">包名 1--split--包名 2...</string>

</resources>
```

其中, “包名 x”是你希望出现在下拉菜单上的应用包名, 包名与包名之间用"--split--"字符串隔离; “aston-xxx”是你的方案名称。

### 4.3. 注意内容

如果 right\_memu\_app\_list 中包含空的内容, 下拉菜单将会生成所有已安装应用的图标。

## 5. 屏幕适配

### 5.1. 分屏窗口大小配置

当更换屏幕或者需要调整 ABC 各区域的大小，可以按照以下方法调整。

A 区宽度配置：更改 frameworks/base/core/res/res/values/dimens.xml 中 navigation\_bar\_width\_splitscreen 的值。

```
<dimen name="navigation_bar_width_splitscreen">200dp</dimen>
```

B 区宽度配置：更改 frameworks/base/core/res/res/values/dimens.xml 中 left\_window\_width 的值。

```
<dimen name="left_window_width">200dp</dimen>
```

C 区大小根据屏幕宽度以及 AB 区宽度自动算出，不需要额外配置。

## 6. 分屏开关配置

### 6.1.编译时的配置项

分屏功能的开关配置项位于 `android\device\softwinner\magton-p1\magton_p1.mk` 中。配置内容如下：

```
# ##### SPLIT SCREEN CONFIG BEGIN #####  
  
PRODUCT_PROPERTY_OVERRIDES += \  
    persist.sys.splitscreen=true  
  
# ##### SPLIT SCREEN CONFIG END #####
```

当 `persist.sys.splitscreen` 的值为 `true` 时，打开分屏模式，当 `persist.sys.splitscreen` 的值为 `false` 时，关闭分屏功能，使用 Android 原生的显示模式。注意，更改该配置值后，应当执行"`make installclean`"命令后，再编译固件。

### 6.2.系统运行时的配置项

系统运行时，可以通过 `setprop` 命令，修改 `persist.sys.splitscreen` 的值去开关分屏功能，步骤如下：

#### 1.打开分屏功能

```
setprop persist.sys.splitscreen true  
  
reboot
```

#### 2.关闭分屏功能

```
setprop persist.sys.splitscreen false  
  
reboot
```

## 7. 系统默认应用配置

### 7.1. 数据存储位置

系统默认有三个应用，分别是：行车记录仪应用，launcher 应用，右窗口默认显示应用。这三个应用的包名配置，在系统运行时存储在 settings.db 数据库的 system 表中。数据类型是 String，包名和 Activity 名以“--split--”字符串隔离，字段名分别如下：

表 4-1

应用名称	数据库字段名	数据库内容
行车记录仪	def_car_video_recorder_name	com.android.camera2--split--com.android.camera.v66.CarDrivingRecorder
launcher	def_launcher_name	com.softwinner.launcher--split--com.softwinner.launcher.Launcher
右窗口默认应用	def_right_window_app_name	com.example.test1--split--com.example.test1.MainActivity

上述表格中，数据库内容是可配置内容。

### 7.2. 配置方法

配置方法同 3.2 节，在 overlay 中需要加入以下内容：

```
<string name="def_launcher_name"
translatable="false">com.softwinner.launcher--split--com.softwinner.launcher.Launcher</string>
<string name="def_right_window_app_name"
translatable="false">com.example.test1--split--com.example.test1.MainActivity</string>
<string name="def_car_video_recorder_name"
translatable="false">com.android.camera2--split--com.android.camera.v66.CarDrivingRecorder</string>
>
```

分隔字符串“--split--”之前的内容是包名，之后的内容是启动的主 Activity。

## 8. 右窗口应用开发注意事项

1. 能在右窗口上运行的应用，不是通用的应用程序，而是针对右窗口的特性进行适配过的应用；
2. 应用的界面能实时自适应窗口大小；
3. 右窗口的应用不能有触发进入全屏的条件；
4. 由于显示区域有限，应用界面不能过于复杂和紧凑；
5. 应用自带返回按钮或界面切换按钮，但应用不能退出自己；
6. 应用的界面，尽量在一个 Activity 中实现；
7. 应用知道自己（或由它启动的应用）需要在右窗口启动的行为时，应该在 Intent 中加入右窗口启动的标志；
8. 应用要做好随时被关闭的准备，在 onPause 和 onStop 中做好处理；
9. 同一个应用的不同 Activity，不要使用 singleInstance 启动模式，慎重使用 singleTask 启动模式。

## 9. 调试方法

### 9.1. 查看左右窗口任务栈的内容及窗口区域

命令：

```
am stack boxes
```

输出内容样例：

```
root@aston-p1:/ # am stack boxes

Box id=4 weight=0.0 vertical=false bounds=[975,0][1600,480]

Stack=

  Stack id=4 bounds=[975,0][1600,480]

    taskId=6: com.android.settings/com.android.settings.Settings

Box id=3 weight=0.0 vertical=false bounds=[175,0][975,480]

Stack=

  Stack id=3 bounds=[175,0][975,480]

    taskId=5: com.android.gallery3d/com.android.gallery3d.app.Gallery

Box id=0 weight=0.0 vertical=false bounds=[175,0][1600,480]

Stack=

  Stack id=0 bounds=[175,0][1600,480]

    taskId=4: com.android.launcher/com.android.launcher2.Launcher
```

其中 Box id 是 ActivityStack Id，bounds 为窗口大小，ActivityStack Id 为 0 的窗口，是 launcher 的窗口，另外两个窗口分别为左窗口和有窗口。需要根据后面的命令确认。

### 9.2. 查看左右窗口的状态

左右窗口的 StackId、window 尺寸、window 状态为调试可测内容。命令：

```
dumpsys activity split
```

输出结果如下：

```
root@aston-p1:/ # dumpsys activity split

split window info:
```

Left window:

StackId=2 WindowStatus=0

StackId=0 WindowStatus=0

StackId=7 WindowStatus=0

StackId=9 WindowStatus=0

StackId=6 WindowStatus=0

Right window:

StackId=13 WindowStatus=0

FullWindowRect=null

Left window 和 Right window 分别列出了左窗口和右窗口的 ActivityStack id 及窗口状态。WindowStatus 显示值所对应的状态，请参考表 2-1.

### 9.3. 打开分屏调试 log

分屏调试 log 的配置位于：

```
frameworks/base/services/java/com/android/server/am/ActivityManagerService.java
```

将 ActivityManagerService.java 的 SPLIT\_SCREEN\_DEBUG 设置为 true 打开分屏调试，将 SPLIT\_SCREEN\_DEBUG 设置为 false，关闭分屏调试。



## **Declaration**

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.