添加自动亮度调节到QuickSetting

（1）问题描述：

在下拉状态栏中的QuickSetting添加自动亮度调节快捷设置按钮。

（2）芯片：

MTK6737

（3）系统版本：

android M

（4）标签：

android 6.0

（5）附件：



（6）解决办法：

1. Android M解决方法：

a.在frameworks/base/packages/SystemUI/src/com/android/systemui/qs/tiles/目录下创建自动亮度调节tile类：

AutoAdjustBrightnessTile.java

b.由于设置自动亮度调节是通过：Settings.System.putInt(mContext.getContentResolver(), SCREEN\_BRIGHTNESS\_MODE,

enabled ? SCREEN\_BRIGHTNESS\_MODE\_AUTOMATIC : SCREEN\_BRIGHTNESS\_MODE\_MANUAL);

来设置的，所以需要一个状态监听器：

SystemSetting.java

c.在frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/phone/QSTileHost.java类中的createTile(String tileSpec)方法添加如下红色字体代码：

protected QSTile<?> createTile(String tileSpec) {

IQuickSettingsPlugin quickSettingsPlugin = PluginFactory

.getQuickSettingsPlugin(mContext);

if (tileSpec.equals("wifi")) return new WifiTile(this);

……

/// M: Add extra tiles in quicksetting @{

else if (tileSpec.equals("hotknot") && SIMHelper.isMtkHotKnotSupport())

return new HotKnotTile(this);

else if (tileSpec.equals("audioprofile") && SIMHelper.isMtkAudioProfilesSupport())

return new AudioProfileTile(this);

// /@}

///Xunhu: add automatically adjust brightness title in quicksetting at 2016-10-24 18:17:14 by QinTuanye{{&&

///Description: 添加自动调节亮度功能到快捷设置栏中

else if (tileSpec.equals("autobrightness") && true)

return new AutoAdjustBrightnessTile(this);

///&&}}

/// M: add DataConnection in quicksetting @{

else if (tileSpec.equals("dataconnection") && !SIMHelper.isWifiOnlyDevice())

return new MobileDataTile(this);

……

else if (tileSpec.startsWith(IntentTile.PREFIX)) return IntentTile.create(this,tileSpec);

else throw new IllegalArgumentException("Bad tile spec: " + tileSpec);

}

d.修改frameworks/base/packages/SystemUI/src/com/android/systemui/qs/QSTile.java类中的getDrawable(Context context)方法为（如果添加的tile使用的Drawable都是<vector/>，则可以不修改该文件）：

@Override

public Drawable getDrawable(Context context) {

// workaround: get a clean state for every new AVD

///Xunhu: add automatically adjust brightness title in quicksetting at 2016-10-24 18:17:14 by QinTuanye{{&&

///Description: 添加自动调节亮度功能到快捷设置栏中

/\*

final AnimatedVectorDrawable d = (AnimatedVectorDrawable) context.getDrawable(mResId)

.getConstantState().newDrawable();

d.start();

if (mAllowAnimation) {

mAllowAnimation = false;

} else {

d.stop(); // skip directly to end state

}

return d;

\*/

Drawable drawable = context.getDrawable(mResId).getConstantState().newDrawable();

if (drawable instanceof AnimatedVectorDrawable) {

final AnimatedVectorDrawable d = (AnimatedVectorDrawable) context.getDrawable(mResId)

.getConstantState().newDrawable();

d.start();

if (mAllowAnimation) {

mAllowAnimation = false;

} else {

d.stop(); // skip directly to end state

}

return d;

} else {

return drawable;

}

///&&}}

}

e.修改frameworks/base/packages/SystemUI/src/com/android/systemui/tuner/QsTuner.java类中的如下方法：

private static int getLabelResource(String spec) {

if (spec.equals("wifi")) return R.string.quick\_settings\_wifi\_label;

……

// M: Add extra tiles @{

else if (spec.equals("audioprofile")) return R.string.audio\_profile;

else if (spec.equals("hotknot")) return R.string.quick\_settings\_hotknot\_label;

// @}

///Xunhu: add automatically adjust brightness title in quicksetting at 2016-10-24 18:17:14 by QinTuanye{{&&

///Description: 添加自动调节亮度功能到快捷设置栏中

else if (spec.equals("autobrightness")) return R.string.quick\_settings\_brightness\_label;

///&&}}

return 0;

}

下面方法是由于在frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/phone/QSTileHost.java类中的loadTileSpecs(String tileList)方法中使用插件修改了默认标题数量、内容和顺序导致与真实默认的值不一致，从而导致数组下标越界的可能，因此在这里也使用插件对默认值进行处理，以使其一致。

public void showAddDialog() {

List<String> tiles = mTileSpecs;

int numBroadcast = 0;

for (int i = 0; i < tiles.size(); i++) {

if (tiles.get(i).startsWith(IntentTile.PREFIX)) {

numBroadcast++;

}

}

///Xunhu: add automatically adjust brightness title in quicksetting at 2016-10-24 18:17:14 by QinTuanye{{&&

///Description: 添加自动调节亮度功能到快捷设置栏中

// M: add extra tiles

/\*

String[] defaults = (getContext().getString(

R.string.quick\_settings\_tiles\_default)

+ "," + getContext().getString(

R.string.quick\_settings\_tiles\_extra)).split(",");

\*/

IQuickSettingsPlugin quickSettingsPlugin = PluginFactory.getQuickSettingsPlugin(getContext());

String defaultTileList = quickSettingsPlugin.customizeQuickSettingsTileOrder(getContext().getString(

R.string.quick\_settings\_tiles\_default)

+ "," + getContext().getString(

R.string.quick\_settings\_tiles\_extra));

String[] defaults = defaultTileList.split(",");

///&&}}

final String[] available = new String[defaults.length + 1

- (tiles.size() - numBroadcast)];

……

}

f. 由于在frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/phone/QSTileHost.java类中的loadTileSpecs(String tileList)方法中使用插件对默认值进行了处理。插件在进行处理时直接使用插件中的值对默认值进行替换，因此需要修改插件中的默认值，使其存在自动亮度调节tile。在vendor/mediatek/proprietary/operator/OP02/packages/apps/Plugins/res/values/config.xml中的添加如下红色字体代码：

<!-- The default tiles to display in QuickSettings for Single Sim-->

<string name="quick\_settings\_tiles\_Single\_Sim\_default" translatable="false">

wifi,bt,inversion,dnd,cell,dataconnection,airplane,rotation,flashlight,location,autobrightness,cast,hotspot,hotknot,audioprofile

</string>

<!-- The default tiles to display in QuickSettings for Dual Sim-->

<string name="quick\_settings\_tiles\_Dual\_Sim\_default" translatable="false">

wifi,bt,inversion,dnd,cell,dataconnection,airplane,rotation,flashlight,location,autobrightness,cast,hotspot,hotknot,audioprofile

</string>

注意：插件的位置可能不是这个，可以通过打印frameworks/base/packages/SystemUI/src/com/android/systemui/statusbar/phone/QSTileHost.java类中loadTileSpecs(String tileList)方法中的quickSettingsPlugin变量的完整类名来获取实际使用的插件。